DOCUMENT RESUME

ED 262 761 IR 011 839

AUTHOR Carey, Doris; Carey, Regan
TITLE Microcomputers in Education: A Self-Paced Orientation.
INSTITUTION International Council for Computers in Education, Eugene, Oreg.
PUB DATE Sep 84
NOTE 96p.; Manual requires Bank Street Writer, Know Your Apple IIe, Apple Presents. . . Apple.
AVAILABLE FROM International Council for Computers in Education, 1787 Agate Street, Eugene, OR 97403-1923 ($10.00).
PUB TYPE Guides - Classroom Use - Materials (For Learner) (051) -- Collected Works - Conference Proceedings (021)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS Computer Assisted Instruction; *Computer Literacy; *Computer Software; *Individualized Instruction; *Learning Modules; *Microcomputers; Pacing; Teaching Methods; Word Processing
IDENTIFIERS Apple IIe; *Computer Uses In Education

ABSTRACT

Designed to serve as a self-paced computer course for education students with no experience using microcomputers, this manual contains instructions for operating an Apple IIe microcomputer, its introductory software, and Bank Street Writer, using the DOS 3.3 System Master. The lessons, which contain illustrations and sample screens, include information on the Apple IIe system; using the Apple IIe keyboard; initializing a diskette on the Apple IIe; entering, saving, and loading programs; computer assisted instruction, and word processing. Also included are selected readings: (1) "Computer Literacy: Talking and Doing" (Dave Moursund); (2) "I Don't Know" (Dave Moursund); (3) "Informational Technology and Its Impact on American Education" (Office of Technology Assessment); (4) "Toward Improving Learning Strategies and Personal Adjustment with Computers" (Kenneth Ryba and James Chapman); (5) "Toward a Computer Literate Society--An Elementary School Responsibility" (Kay Walla and Virginia Brubaker); (6) "The Intelligence Amplifier: An Essay on the Design of Computer-Based Curricula" (Michael Moshell); (7) "Should Young Children Work with Microcomputers--Logo before Lego?" (B. J. Barnes and Shirley Hill); (8) "Microworlds" (Glen Bull); (9) "Creating a Logo Environment" (Tim Riordon); (10) "Turtle Fever" (Rick Billstein); (11) "Keyboarding--A Must for Tomorrow's World" (Evelyn Kisner); (12) "Sex Equity: Increasing Girls' Use of Computers" (M. E. Lockheed and S. B. Frakt) and (13) "A Nod to the Novice--#1-6" (Bob Skapura). (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Microcomputers in Education

A Self-Paced Orientation

Doris Carey and Regan Carey

Graphics: Bank Street Writer
Know Your Apple IIe
Apple Presents ... Apple

BEST COPY AVAILABLE

Publications

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
David Morsund

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

About the Authors

Doris Carey is working on a doctorate in computer education and a masters in computer science at the University of Oregon. She has taught mathematics for 11 years, seven of which were in British Columbia. She has also taught computing in high school and to adults. Her particular interest areas are preservice and inservice teacher education.

Regan Carey is in the computer education doctoral program at the University of Oregon. He has taught social studies at the secondary level in British Columbia. In addition he has conducted workshops in instructional computing and taught programming to adults. Regan is very interested in historical simulation games.

Doris and Regan have both worked for ICCE as Software Review Editor and Book Review Editor, respectively, for *The Computing Teacher*. They both travel extensively to present computer education workshops.

This booklet is published by the International Council for Computers in Education, a non-profit professional organization. ICCE is dedicated to improving educational uses of computers and to helping both students and teachers become more computer literate. ICCE publishes *The Computing Teacher*, a journal for teachers and for teacher educators. It also publishes over 15 booklets of interest to educators.

The booklet prices given below are for **prepaid** orders. On other orders a $2.50 processing charge will be added.

QUANTITY	PRICE (U.S. $)
1-4 copies	$10.00
5-9 copies	$ 9.00
10-99 copies	$ 8.00
100 + copies	$ 7.00

Please place your orders with:

International Council for Computers in Education
University of Oregon
1787 Agate St.
Eugene, OR 97403-1923
(503) 686-4414, 686-4429

Copyright © ICCE 1984
(First Printing, September 1984)

Apple IIe is a registered trademark of Apple, Incorporated.

Table of Contents

Introduction 2

1. The Apple //e System 3

2. Using the Apple //e Keyboard 7

3. Initializing a Diskette on the Apple //e..................... 11

4. Entering, Saving, and Loading Programs 15

5. Computer Assisted Instruction 19

6. Word Processing 22

READINGS

Computer Literacy: Talking and Doing *Dave Moursund* 33

I Don't Know *Dave Moursund* 34

Informational Technology and Its Impact on American Education *Office of Technology Assessment* 35

Toward Improving Learning Strategies and Personal Adjustment with Computers *Kenneth Ryba* and *James Chapman* 46

Toward a Computer Literate Society--An Elementary School Responsibility *Kay Walla* and *Virginia Brubaker* 52

The Intelligence Amplifier: An Essay on the Design of Computer-Based Curricula *Michael Moshell* 55

Should Young Children Work with Microcomputers--Logo Before Lego? *B.J. Barnes* and *Shirley Hill* 60

Microworlds *Glen Bull* 64

Creating a Logo Environment *Tim Riordon* 66

Turtle Fever *Rick Billstein* 71

Keyboarding--A Must for Tomorrow's World *Evelyn Kisner* 74

Sex Equity: Increasing Girls' Use of Computers *M. E. Lockheed* and *S. B. Frakt* 76

A Nod to the Novice -- #1-6 *Bob Skapura* 79

BEST COPY AVAILABL

Preface

Orientation to Microcomputers in Education was developed at the University of Oregon to serve as a self-paced course for education students with no previous experience on microcomputers. It is offered as a stand-alone course, or as a prerequisite to an introductory level Computers in Education survey course. The course should take four or five weeks for a student to complete. The version presented here is the result of four terms of adjusting and editing the course to minimize problems for both students and instructor.

The course requires that students have reasonably open access to an Apple //e system and occasional access to a system equipped with a printer. In addition, certain commercial software is necessary. The required software is:

Bank Street Writer by Scholastic or Broderbund
Know Your Apple //e by Muse
Apple Presents .... Apple by Apple
DOS 3.3 System Master by Apple

It is also recommended that students have access to as wide a variety of educational software as possible. While additional software is not required by the course, students should be encouraged to explore it to provide some depth to their knowledge.

Over the past year, we have found that *Orientation* requires a minimum of instructional support. The instructor should be familiar with the course material, and be available on a regular basis to help students with any problems that arise. There is no need for a regular class meeting.

Introduction

This booklet contains all the instructions you will need to follow the course Orientation to Computers in Education. Much of the work you will be required to do will consist of "hands-on" computer activities. This means that you will do much of the course while sitting at a microcomputer. All that is needed, with the exception of the computer software, has been provided here, and you need only follow the instructions. It is highly recommended that you spend the first little while looking through this handbook.

The readings in this book have been selected to give you a survey of some of the current ideas about the use of computers in education. You should work through the readings at home, between "hands on" sessions in the computer lab.

You will be working on the Apple //e microcomputer system. This is a brand of microcomputer which is found in most of computer-using American schools. Much of the existing computing materials for education have been prepared for use with the Apple microcomputer. Many of the skills that you will learn can be transferred to other brands of microcomputers. It is hoped that this course will help you understand some of the concepts behind computer use and applications.

Lesson 1.

The Apple //e System

This section will take you on a guided tour of the computer system. The machinery itself is called the *hardware*. This time, we will be using the diskette called *KNOW YOUR APPLE //e*.

Before you turn it on, have a look at the computer. The main part of the computer where all the work is done is called the Central Processing Unit. In this particular case, the "inner workings" of the Apple are stored in the unit that contains the keyboard.

You may want to stop and think about the similarities between computer use and typewriter use, especially with regard to use by young children. In computer education, we are concerned about the age at which children should or should not learn touch typing. You will probably notice that the touch is quite soft on a microcomputer, compared even to an electric typewriter. You will probably find it easier to keep your eye on the screen than on the keyboard when typing. Again the issue of standard typing skills for children looms as a concern for educators: should children be taught proper typing skills, and, if so, by whom and when?

Directly in front of you, you will see a device much like a television. This is a monitor, and the main difference between a monitor and a TV is that the monitor is equipped to pick up its signals directly from the computer, rather than from the air (via an antenna or other device). The result is that the monitor offers a sharp clear picture. This is especially important when you are workings with letters, numbers, and other small symbols, or when lines need to be fine. You will form your own opinions when you have contrasted color monitors, monochrome (single color) monitors which can be black/white, black/green, or black/amber, and regular TV sets. For a great variety of prices, you can obtain greater or lesser quality in sharpness and color.

The next item to note is the rectangular box or boxes located beside or on top of the computer. Each horizontal slot (you may have one or two with the system that you are working on) is a disk drive and its function is much like that of a tape deck in your home stereo. Once a disk is inserted, contact is made and the disk is "read." This is much like the insertion of a record onto a turntable, where the needle descends on the record and begins to "read" the music. Of course, some disks are blank. If you have a tape drive on your stereo and you insert a blank cassette tape, you will get no sound. The disk you brought with you, if it is brand new, is blank. You will learn how to fill it soon.

The last device that you should look at is the printer. There may be a printer near you, or it may be necessary to walk over to a printer which is linked to another microcomputer to have a look. You will again see similarities with the typewriter, except that this device definitely lacks a keyboard. Paper is being fed in on continuous sheets which are perforated for easy tearing. You will

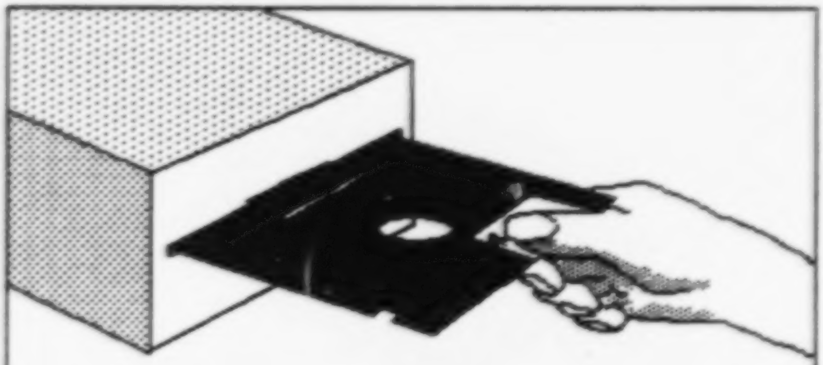
BEST COPY AVAILABLE

get more advice on this later. What you need to know is that it is not necessary to have a printer by your side at all times in order to use a computer. You can "record" any work on your diskette, then carry the diskette over to a computer which is adjacent to a printer when you need a printout. It takes only a few minutes to print out a ten-page document.

Before we begin to use the computer, you should examine the diskettes. Diskettes are delicate, subject to damage by excess heat, cold, magnetism, static electricity, fingerprints, bending, and many other types of careless handling. You may want to think of your disk as being similar to a favorite LP record. When in doubt, handle cautiously.

Now we are ready to begin!

1. Remove the *Know Your Apple //e* diskette from the envelope by grasping it with your right hand. Your thumb should be on the label. If you continue to hold the diskette by this corner, face up, then all should go well. Open the door to the disk drive by flipping it up. If you are working on a two drive system, use Disk Drive #1. Next, gently slide in the diskette.



Keep your thumb on the label.

If it does not slide in easily, do not force it. Just continue to guide it in. Flip the disk drive door down again.

2. Flip the switch on the front of the monitor to ON. Nothing will happen, except that the monitor light will come on. The light will probably be red or a green, depending upon the type of monitor that you are working on. Some monitors don't even have a light to show that they are on. If this is the case, don't worry.

3. Reach behind the computer with your left hand. You may encounter a handful of wires, but persist until you find a small switch just around the back left corner (about one and a half inches in). Turn this rocker switch on.

As soon as you do this, the green light on the computer will come on, the disk drive will begin to rattle and whir, and eventually something will appear on the screen. Notice that the tiny red light on the disk drive is on while the disk drive is whirring. Wait until the disk drive has stopped and the little red light is off before you continue.

The *Know Your Apple //e* title page should appear, with the program name in large letters. At the bottom of the screen you will see a line that says:

Press any key to continue.

This is called a *prompt*, and it is there to tell you what to do next. Do what it says and press any key ...

The next thing that you will see is the "menu." This is like a table of contents. Not all disks have a menu, but this program has a very nice one. Unlike most menus, which simply use words, this program presents you with a picture of a typical Apple //e system (with the old style single disk drive). Again you will see prompts at the bottom of the page. In this case, pressing the space bar will move the little arrow through the list of computer system parts that you may investigate in the tutorial.

Try pressing the space bar a few times.

When you press the key marked RETURN, the part of the program that is indicated by the arrow will be loaded from the disk into the computer.

4. The next step is to go through the contents of *Know Your Apple //e* on your own. A summary of the commands that you can use is given below.

You needn't go through the entire diskette at one sitting. If you must leave before you have seen all the sections, simply remove the diskette from the disk drive, return it to its jacket, then turn off both the monitor and the computer.

You may wish to take notes as you go along. Some of the information you will obtain will help you later. Be sure to go through all sections of the menu. We will assume that you have been through the entire contents of the diskette in future lessons.

SUMMARY OF KNOW YOUR APPLE //e COMMANDS:

When the menu is displayed:

Press:	**SPACEBAR**	to move the "selector" arrow to the lesson of your choice.
	RETURN	to run the selected lesson.

At any time from within a lesson:

Press:	**RETURN**	to proceed to the next frame.
	-	to review the previous frames.
	CTRL and **C**	together to interrupt the current lesson and display the menu.

At the end of a lesson:

Press:	**R**	to repeat the lesson.
	M	to display the menu and choose another lesson.

Lesson 2.

Using the Apple Keyboard

Before you begin, make sure that you have the diskette labelled *Apple Presents ... Apple*. You should be sitting in front of an Apple //e microcomputer.

"Boot" the *Apple Presents ... Apple* disk.

Go through all the sections of this diskette as you did for *KNOW YOUR APPLE //e*. Again, you are welcome to take notes. You will probably add many new words to your vocabulary. It will be helpful to note these words as you go along.

BEST COPY AVAILABL

SELF-QUIZ #1

After you have finished looking at *APPLE PRESENTS ... APPLE* and *KNOW YOUR APPLE //e* you may wish to check your new knowledge. Here are a few questions for you. The answers are on the page following the quiz. This quiz is intended to help you review. There is no advantage to peeking at the answers ahead of time.

If the answers don't make sense to you, you should plan to visit the instructor for this course during office hours to clarify your understanding.

1. The flashing square of light which marks your place on the screen is called
   a) the carriage return.
   b) the cursor.
   c) the monitor.
   d) the control.

2. Diskettes are sensitive to dust, magnetic fields, excessive heat and cold, and rough handling. In addition, diskette users should not
   a) touch the exposed surface of the diskette.
   b) eat solid foods at least 24 hours before touching a diskette.
   c) store diskettes in locked cupboards.
   d) have conversations with blank diskettes.

3. In order to get music from a stereo system, the user must insert a record on the turntable or a cassette in the tape drive. In order to get a program loaded into a computer, the user must insert a diskette in the
   a) central processing unit.
   b) monitor.
   c) printer.
   d) disk drive.

4. In order to use a computer, it is necessary to
   a) know how to program a computer.
   b) take a course in computer science.
   c) own your own computer.
   d) know how to turn the computer on.

5. When you have finished typving an entry in response to a question, you should indicate that you are through by
   a) pressing the RETURN key.
   b) waiting 30 seconds.
   c) pressing the ESC key.
   d) holding down the SHIFT key.

6. In order to view the list of programs stored on a particular Apple //e diskette, the user should lock down the CAPS LOCK key and type the word
    - a) MENU.
    - b) DISPLAY.
    - c) LIST.
    - d) CATALOG.

7. The main difference between a computer keyboard and a typewriter keyboard is that
    - a) a computer keyboard has the letters arranged differently.
    - b) a computer keyboard has a few extra keys.
    - c) you can damage the computer if you hit keys that are unfamiliar.
    - d) computers can only type capital letters.

8. While a user is running a computer program, it is NOT a good idea to
    - a) smile.
    - b) stop and think.
    - c) press the RESET key.
    - d) look at the monitor while typing.

9. A "user-friendly" program is one which
    - a) runs on a variety of computers.
    - b) uses color, graphics and sound at all times.
    - c) does not require any user response.
    - d) gives clear, simple directions which are not ambiguous.

10. The pictures and drawings that are displayed on the screen can be made up of blocks or fine dots. These choices are called
    - a) low resolution or high resolution.
    - b) pixels or dots.
    - c) graphics or text.
    - d) hardware or software.

Answers to Self-Quiz #1

1. *b* - the flashing square of light is called a *cursor.*

2. *a* - touching the exposed surface of a diskette can leave oil from your finger on it. This can interfere with the mechanism that reads the magnetic fields on the diskette.

3. *d* - this one is pretty obvious!

4. *d* - many programs that allow you to use a computer require nothing more than the insertion of the diskette and the flipping of the power switch. Using a telephone is more complicated!

5. *a* - the use of the word RETURN comes from the old days of the manual typewriter. Once a full line was typed, the roller carriage and the paper had to be returned to the leftmost position, creating the term *carriage RETURN.*

6. *d* - the names of the programs which are stored on a disk are accessed by the DOS command, CATALOG.

7. *b* - because the microcomputer has more capabilities than a typewriter, there is a necessity for a few extra keys, such as ESCape, ConTRL, and RESET.

8. *c* - the RESET key literally "resets" all values in the computer's memory to zero and stops the running of a program. This has obvious detrimental effects on a running program!

9. *d* - a computer program should be easy to run. If it is not, then the *user* is not being treated in a *friendly* fashion.

10. *a* - a picture that is made up of blocks is called *low resolution*, while a picture which is constructed from small dots is called *high resolution.*

Lesson 3.

Initializing a Diskette on the Apple //e

You've probably noticed that each time we've turned the computer on, the disk drive was activated, and there was little for the user (you) to do except respond to the instructions on the screen. You may be curious about what would happen if you did not have a diskette. Let's investigate.

First, check out a copy of the SYSTEM MASTER diskette and have a *blank* disk ready. The disk can be a brand new disk or a diskette that has been used before. IMPORTANT: When you initialize a diskette, all programs and information previously stored on the disk is destroyed. If you are going to initialize a diskette that you or someone else has already used, be certain that there is nothing stored on the diskette that you want to keep!

Make sure the door to the disk drive is flipped up. This will be easier on the equipment. Next, turn on the monitor and the computer. The disk drive whirs and spins, but it never stops!! To stop the disk drive, hold down the CONTROL key and press the RESET key SIMULTANEOUSLY. Now the disk drive has stopped whirring. Unfortunately, now we do not have any disk interaction, and nothing to do.

Let's experiment with a disk and see exactly what happens when the disk drive is activated. Turn the computer off, and insert the disk labelled SYSTEM MASTER. If it is a two-drive system, use Drive #1. Make sure the disk drive door is shut and turn on the computer (is the monitor still on?).

The process of turning on the computer to cause a disk to begin loading is called "booting" a disk. This expression comes from the phrase "pulling up by your own bootstraps" and signifies roughly a process whereby a system gets itself going all by itself.

Several things happened when you booted this disk. First, the computer searched the disk for a set of instructions. Any set of instructions is called a program (more about this later). The first set of instructions it found actually helped it to interact with the disk drive. This set of instructions caused the computer to keep searching for more instructions on the disk, so the second effect of booting the disk was to load a version of BASIC (called Integer BASIC) into the computer's memory. In this particular case, we won't be needing Integer BASIC, but the computer loads it automatically, just in case. When everything is ready, the Applesoft BASIC prompt] is displayed.

Let's see how the instructions work.

Disk drives and computers need very specific instructions to make them work. Computers have some instructions which are permanently fixed in their memories. They are part of the computer's permanent memory. Such instructions cause the computer to respond to key presses, to display letters and symbols on the screen, and many other functions. Unfortunately, disk drives do not have this type of built in instruction, and the computer must have specific instructions in order to run the disk drive.

This set of instructions is called the Disk Operating System, known as the DOS. It is not part of the computer's permanent collection of instructions. That is why the DOS must be loaded into the computer each time you wish to use the disk drive. Now this set of instructions, the DOS, is the first thing that gets transferred into the computer's temporary memory when the computer is turned on.

Of course, this requires that that particular list of instructions (DOS) actually can be found on the disk you inserted into the disk drive. Since our Apple CAI disk does contain these instructions, you actually accomplished this transfer by booting the disk. This is called "booting DOS". As you've probably guessed by now, your new disk does not contain these instructions, and we must make sure that it does.

Let's work on your vocabulary for a little while. That permanent memory that we talked about cannot be tampered with. We can only access it or "read" from it. It is known as "Read Only" Memory, and is abbreviated as ROM. The other type of memory that temporarily stores lists of instructions such as DOS and many other lists of instructions, is easily accessed in arbitrary spots. It is known as "Random Access" Memory, abbreviated RAM.

You have probably noticed that we have referred to lists of instructions and programs interchangeably so far in the text. That is because a program IS simply a list of instructions for the computer.

Think of it this way. If you give a child instructions to make a sandwich, you might want to pause after each instruction and wait while s/he carries it out. On the other hand, an older child might be able to wait while you give out all the instructions (i.e., get the bread from the breadbox and the peanut butter from the pantry, get the butter from the refrigerator, take a knife out of the drawer, etc.) then carry out all the instructions at once after you have finished talking.

A list of instructions that is not carried out as you go along is called a program. You can probably see where the word "memory" is used. It is simply a storage capacity, and in the case of the computer, includes a capability for storing electronic impulses or signals that can be interpreted and put to use. On one hand, it is desirable to use words in common usage to describe computer functions (the volume of new words is onerous enough as it is), but on the other hand it is easy to attribute human characteristics (like memory) to a boxful of electronic devices because of the human-like descriptors.

Let's put this new information to good use. Our next objective is to prepare your diskette for use. We will have to go through a process called "initializing a diskette." Notice that the words disk and diskette are both used to describe the same item.

The first thing that we must do is to boot the SYSTEM MASTER disk. If you have not turned the computer off since booting the SYSTEM MASTER a few paragraphs ago, then you don't have to do it again. Booting the SYSTEM MASTER will put DOS (plus a few other things) into the computer's memory. We do not want to use the program that the process of booting loaded into memory, so we'll clear the computer's memory by typing

```
NEW
```

and pressing RETURN. (nothing appears on the screen)
(This does not clear DOS out of memory, just the program.)

Now we must put a new program into memory. *Be sure to press RETURN after every line that you type in.* Type in the following statements:

```
10 REM  HELLO PROGRAM
20 REM  STORAGE AND WORK DISK
30 REM  ORIENTATION TO MICROS COURSE
40 END
```

You may wish to add an extra line or two, stating your name or any other information that you consider pertinent. Be absolutely sure that you put the word REM immediately after the line number and before your comments. Also, be sure that the highest numbered statement in your program is the END statement. e.g. a sample line would be:

```
35 REM  THIS DISK BELONGS TO REGAN CAREY
```

Now, take the SYSTEM MASTER disk out of the disk drive and put it into its envelope. Put your new disk into the disk drive and close the door.

Type:

```
INIT HELLO  ( and press RETURN)
```

The disk drive will now whir and click for a minute or two. It

BEST COPY AVAILABLE

is arranging the storage space on the disk to make it suitable to use with an Apple //e. It also puts DOS onto the disk.

Now type:

CATALOG (and press RETURN)

If your disk has been properly initialized; the disk will whir for a couple of seconds and then the screen will show a "catalog" of the programs on your disk. Of course, at this time, you have only one program, called HELLO, on the disk. Using the command CATALOG is a very useful way to remember what programs you have stored on a disk.

You can load your HELLO program by typing:

LOAD HELLO (and pressing RETURN)

Type LIST (and press RETURN) to list the lines in your program.

Your disk has also been prepared to accept new programs. We'll actually be saving programs on this disk in a coming lesson.

Lesson 4.

Entering, Saving, and Loading Programs

The best way to see what a program is and how it works is to write one yourself. Of course, this is not a programming class, so you will not be required to actually design the program itself. However, it is not difficult to type in a short program which someone else has written for you, so let's do that now. When you turn on the computer, you will not have a disk in the drive. This means that it will start to look for a disk, and will keep working at it until you interrupt the process. Do this by pressing the keys marked CTRL and RESET simultaneously. As you learned earlier, the CTRL key is on the left side of the keyboard and the RESET key is located on the upper right side of the keyboard. The disk drive will now stop whirring. The two keys are placed far apart and must be both be pressed at the same time in order to prevent an accidental RESET.

You will notice that a square bracket has appeared on the screen. This is known as a "prompt," and this particular prompt tells you that the Applesoft BASIC language is ready to be used. This BASIC only recognizes capital letters, so depress the CAPS LOCK key and leave it that way for the remainder of this lesson.

Now the bottom of the screen is not a good place to start, so let's try to move that marker up higher. Just type HOME and press RETURN. If you need to type a symbol (such as ?, !, or &) just use the SHIFT key as you normally would.

The program that we will be entering can be SAVED on your disk, just as music or speech can be recorded on a tape. To do this, we must be sure that DOS has been loaded into the computer before we start to type in a program. So far, we have "booted" a disk by simply turning the computer on. What happens if the computer is already on?

Put your diskette in Drive #1.

The computer communicates with the disk drive by means of a special insert (called a "card") located in an add-in slot inside the computer. If you were to look inside the Apple, you would see that the cable from the disk drive(s) plugged into a card in the rear right hand corner of the computer's main board. There are seven slots on the main board of the Apple, numbered from 0 to 6. Usually the disk drive card is located in slot number 6. In order to activate Disk Drive #1, we want the computer to "use" the card in Slot 6. So, we type in:

PR#6 (and press RETURN). *This tells the computer to "print" using slot 6.*

The disk drive will whir and click and the Applesoft prompt (right square bracket) will appear again, with the flashing cursor. DOS has been loaded into the computer's memory now, and it is ready to receive commands that allow you to interact with a disk later.

The program needs to be entered, one instruction at a time. Each instruction will be called a line of the program. Each time you finish typing a line, you should check for spelling or typographical errors. If you find one, use the left arrow key (<--) to back up and correct the error. Once a line is correct, press the RETURN key to indicate that the computer can absorb your entry. You will see that pressing the RETURN key will also cause a carriage return to the left of the next line, just like the carriage return on a typewriter.

Go ahead and type in this program, one line at a time, pressing the RETURN key after each line has been typed correctly.

```
10  FOR X = 1 TO 20
20  PRINT "COMPUTERS ARE VERY FAST"
30  NEXT X
40  END
```

Check again to make sure that each line is exact, with the correct punctuation. Be certain that you have used a zero, not the letter O, where called for. They may look similar, but they are very different to the computer. If you're sure that the program is typed in correctly, type in the word RUN and press the RETURN key.

The screen should display twenty lines of "COMPUTERS ARE VERY FAST." Did you notice that the screen never runs out of room? The process of moving writing upward to make room for more text is called *scrolling*. Scrolling is always an upward movement. Unfortunately, once the screen has scrolled upward, you cannot roll it back down again. However, if you wish to look at the program which you typed in a minute ago, you can call it back. This because the program has been saved in the temporary memory called RAM. To recall your program, type LIST (don't forget to press RETURN!).

Listing a program is very useful. It allows us to make changes, whether to correct errors or to have the program do something quite different. Making changes is called EDITING. Let's edit this program. First, we'll retype line 10 to make the program print 100 lines instead of 20. To do this, simply type

```
10  FOR X = 1 TO 100
```

and press RETURN. Type LIST to see the changes made to the original program. You have probably noticed that the new line 10 has replaced the old line 10. Execute this new program by typing RUN. Did you notice that the screen contents scrolled up as soon as the screen ran out of space for the lines being displayed?

We could do a lot of experimenting with this little program, but what we are really interested in is transferring the program to your new diskette. This is very easy to do. First, you can be creative and decide on a name for your program. The name should be short and simple, such as EXPERIMENT, or CASPER, or SAMPLE. For the sake of clarity, we will use the name DEMO in these instructions, but you can use any name you like.

Now save the program you wrote by typing SAVE DEMO (with your chosen program name instead of DEMO, of course) and pressing RETURN. Wait until the disk drive stops running before you proceed.

Now let's see what happens after a program has been saved on disk. Clear the screen by typing HOME (and pressing RETURN). We want to see if the computer "remembers" our program. Type LIST and press RETURN. You can see that the program was copied onto your diskette, but the computer has not dumped it out of memory. Now remove your disk from the disk drive, and shut off the computer.

Clearly, the computer has now forgotten our little DEMO program. Return the disk to the disk drive and turn on the computer. As far as we know, the short program we saved on disk is not present in the computer's memory. Let's retrieve the program by typing LOAD DEMO (or the name of your program instead of DEMO) and pressing RETURN. The disk drive whirred for a second but nothing happened on the screen. Type LIST and press RETURN.

Ah, here it is! RUN the program (type RUN and press RETURN). All is well so far.

The only additional bit of information you need now is to understand that DEMO still lives on your disk. It would be possible to remove your disk and go load your program into another computer. Like the music on a cassette tape, the program is not erased just because it has been used.

What if we wanted to change DEMO in some way? Let's add a new line to it and see what happens. Type in the following line:

```
     25  PRINT "AND EFFICIENT"
```

and press RETURN. LIST the program and save it again by typing SAVE DEMO (or another appropriate name). Do you remember how to RUN this new program?

Type CATALOG to have a look and see what's on your disk now. Your HELLO program is the first item. This was part of the initializing we did in the last lesson. The second item is your program, but only the last copy has been saved. This is because we used the same name.

If you need to experiment further, feel free to remove your disk from the drive, shut off the computer, and practice loading and running the program until you have a good understanding of the interaction between diskette and computer.

Lesson 5.

Computer-Assisted Instruction Programs

There are many ways in which computers can be used in education. The purpose of this section will be to examine programs which are used to assist in the educational process. There is a growing collection of commercial educational software available. You are encouraged to look over a variety of these instructional programs. They range throughout the entire spectrum of types of CAI.

Drill and Practice

The phrase "drill-and-practice" is used to describe programs which help students practice old skills or recall facts. The first programs written for educational use were mostly drill-and-practice in mathematics. Writing programs for drill-and-practice was easy, so many teachers wrote very elementary, but relatively inadequate programs. Since that time, educators have been asking questions about educational theory, such as the difference between information and knowledge, the appropriateness of instant feedback, how many times we should ask students to guess at answers, and many others.

If you have a collection of educational software available, you should make an effort to view a variety of this type of drill-and-practice program. Try using each program as if you were a student. You may find some design errors or some features that do not make optimum use of the computer. This is especially true in programs that were written in the ancient days of educational microcomputing (about five years ago!)

In summary, the drill-and-practice program does not function as a "teacher", but it allows students to:

- use computers individually to practice skills;
- focus on particular skills that might need practice;
- review facts without the aid of the teacher;
- receive instant feedback on correct or incorrect responses.

Some programs actually do a good job of keeping students' scores and reporting them to the teacher (either on disk or on paper). These programs are much more complex than DRILL AND PRACTICE, and are generally known as CMI programs. CMI stands for Computer-Managed Instruction. This is because the computer acts both as a drill-and-practice device, and as a manager of records.

BEST COPY AVAILABL

Tutorials

Some programs actually teach new facts and skills to the user. We do not need to look very far for such programs. Both *APPLE PRESENTS ... APPLE* and *KNOW YOUR APPLE //e* are examples of tutorial programs. They taught you many new things about the computer that you did not know before. They did not test your knowledge or drill you on facts, but they acted as tutors. Tutorial programs are more difficult to produce than drill-and-practice programs. There are few tutorial programs being produced by educators because they take much longer and require more programming skills. You noticed that your learning was aided by many diagrams and other devices such as sound and animation. These are usually difficult to program. As a result, they are usually produced commercially and are more expensive than other programs. Of course, educators have a great deal to say about what should go into these programs. Again, we want to rely on sound theories of education and learning for the design of programs.

Simulations

Even more complex than tutorial programs are those which simulate events or structures for the purpose of education. The teacher may want graphic representation of complex models such as molecules or the flow of a river. The computer is useful for simulation because it offers animation. The changes and relationships between objects and events can be displayed most easily on a computer screen.

Teachers may also wish to simulate events which require keeping track of great numbers of variables. Examples of such events might be migrations of populations and corresponding changes in food supply, ammunition, manpower, reproduction, etc. Others might include the effects of nuclear explosions or chemical reactions. There are many uses for simulations in a great variety of subject areas.

You can probably see that one computer can be used as a tool for an entire class, and that this opens up many more questions in computer education.

Working all the way through a simulation can be a time consuming process. At first, you may wish to simply review the documentation of some of the simulations that are available in a software collection.

Utilities

Another potential use of the computer in education is as a tool that helps students and teachers carry out tasks in easier and more efficient ways than they do now. For teachers, this might mean programs that allow them to keep and calculate student records quickly and easily. Some teacher utilities are designed to help produce teaching materials, such as crossword puzzles or worksheets. There are also programs available to help prepare and store questions to be used in quiz development.

For students, a utility may be a program that allows them to analyze data obtained in the science lab, or draw graphs from economic information. Other utilities, allow people to easily produce fancy drawings for the computer programs they are writing, or cause printers to produce non-standard lettering. The varieties are endless.

The next section will allow you to experiment with a program that is a utility for both student and teacher. It has been quite a revolutionary addition to the collection of programs used in the classroom. This is the word processing program. A detailed explanation of the word processor and its function is in Lesson 6.

Games:

There are an enormous number of game programs available for microcomputers. In fact, the phenomenon of the video game is one of the major selling points of the home computer (whether we want to admit it, or not). While rationalizations are frequently made about the "educational" value of these games, there some which have definite potential in an educational setting. Such programs as *Master Type* and *Type Attack* teach keyboarding skills, and are among the top selling programs in the country.

Each month, hundreds of new programs flood the market. It is sometimes difficult to keep track of the quality, ease of use, and applicability of these programs. You may find that the five labels described above do not actually fit with a particular program: indeed, sometimes programs cover a variety of areas. You need not be concerned with the classification of programs so much as the facility with the new vocabulary of Computers in Education.

Lesson 6.

Word Processing

Look at Figure 1 below and Figure 2 on the following page. Compare the two letters and see if you can find all the differences.

You can see that it would take a long time to send a personal letter to the parents of each child that is in your class. However, if the letter can be kept in a computer's memory and minor changes made, then it is a much simpler process.

June 10, 1983

Mr. & Mrs. K. Brown
2424 Longview Drive
Jonestown, Oregon

Dear Mr. & Mrs. Brown:

This is to inform you that Timothy's grade five class will be going on a field trip this Friday. We will be spending an hour at the Jonestown Planetarium, then walking to Plato Park to collect moss and grasses for our science experiments.

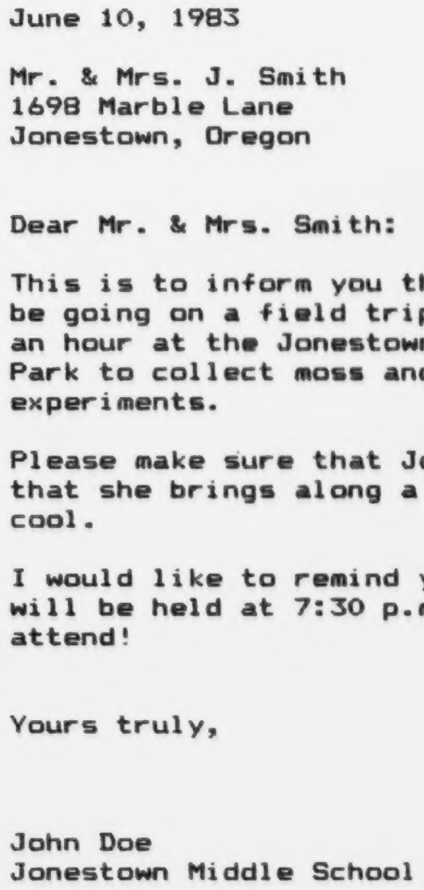
Please make sure that Timothy brings a lunch Friday, and that he brings along a sweater, in case the weather gets cool.

I would like to remind you that our annual Open House night will be held at 7:30 p.m. on June 20th. Please plan to attend!

Yours truly,

John Doe
Jonestown Middle School

Figure 1.



June 10, 1983

Mr. & Mrs. J. Smith
1698 Marble Lane
Jonestown, Oregon

Dear Mr. & Mrs. Smith:

This is to inform you that Joannie's grade five class will be going on a field trip this Friday. We will be spending an hour at the Jonestown Planetarium, then walking to Plato Park to collect moss and grasses for our science experiments.

Please make sure that Joannie brings a lunch Friday, and that she brings along a sweater, in case the weather gets cool.

I would like to remind you that our annual Open House night will be held at 7:30 p.m. on June 20th. Please plan to attend!

Yours truly,

John Doe
Jonestown Middle School

Figure 2.

In this lesson, we will explore the uses of word processors in education. Some modern electronic typewriters have the capacity to remember the last 8 or so letters typed, or even a whole document. When entire documents can be saved in the typewriter's memory, the device is usually called a word processor. This means that the text can be modified and reprinted easily. Names and addresses can be stored for easy addressing of envelopes.

Inexpensive computer programs can now be purchased that simulate these machines. They are called word processing programs, text processors, and other generic names. It is understood that appropriate peripherals, such as printers and disk drives, should also be available.

BEST COPY AVAILABLE

Bank Street Writer is one of the word processing programs that is available to computer users. It is particularly useful in education because it is easy to learn and can be used by children. Figure 3 shows a model of a piece of writing that can easily be edited by the teacher or teacher's aide, then corrected by the child. The benefit of the word processor is that the child need not rewrite the entire composition. Of course, there are many uses for word processors that you may want to investigate, but first, let's see how *Bank Street Writer* works.

The Zoo

Yesturday we went to the zoo we visited the animals and the weather was nice and we had a good time.

Mr. Smith myhomeroom teacher tooks us there in a bus but he did'nt drive the bus.

We saw lions and a tiger before lunch which was very feerce. After we ate, we saw snakes and spiders it was a good thing we saw them, after eating.

Figure 3.

First, boot the Bank Street Writer disk. Make sure the computer is off, insert the disk, and turn on the computer.

Once the Bank Street Writer logo has appeared on the screen, you may put the disk away. It will not be needed as we go along. Insert your own diskette into the drive. From now on, your disk will be used to save your writing.

The first task will be to prepare your diskette to save any writing you have entered. Your disk must be *initialized* to enable you to save the text you enter. The process is similar to the INIT procedure we saw in Lesson 3. You may use a new disk, or an old disk that you aren't using anymore. **Re-initializing your disk will wipe out any previous programs saved on the disk.**

Bank Street Writer helps you to initialize a disk very easily.

First, find the key marked ESC (for ESCape). Notice the list of words which appear at the top of the screen.

Now find the keys with an apple on them. There are two such keys on either side of the space bar. One is full and the other is empty. Continue to press the full apple until the screen words TRANSFER TEXT are highlighted. (You may have noticed that pressing the empty apple moves the highlighting to the left, and the full apple moves the highlighting to the right.)

Now press RETURN, and you will see a new menu at the top of the screen. Use the same procedure to highlight the word INIT. When INIT is highlighted, press RETURN. Notice that the program double-checks to make very sure that you are aware that you will be wiping out any programs already saved on your disk. Type Y for "yes" to indicate that this is okay.

Finally, when your disk is initialized, press ESC twice to return to the text screen.

You may begin to write immediately. Just press any key to begin; don't worry about the instructions on the screen for now. Make sure the CAPS LOCK key has been released or you will not get lower case letters.

Type in a paragraph just as you would on a regular typewriter. If you make a mistake at any time, just back up with the left arrow key (<--) and try again. The SHIFT key works exactly the same way as on a typewriter. If you are an experienced typist, you must resist the urge to press RETURN when your writing reaches the right edge of the screen. Press RETURN only at the end of a paragraph. Press it once to change to a new line, and again to put a blank line on the page before the next paragraph.

Now, let's try to give our paragraph a short title. We will need to move around freely. The power to move around is ours just by changing to a state we will call EDIT MODE.

To enter the edit mode, just press the ESC key. Use the four arrow keys in the lower right corner to move around inside the box on screen. You may not move beyond the end of your paragraph, though. Moving around does not mean changes have been made.

Place the cursor under the first letter of your paragraph. Now press the ESC key again, and you will see that you have re-entered the TEXT mode we were in before. This is where we will insert the title.

First, hold down the CONTROL key and press the C key. Your title will be centered automatically. Type in a title (try to keep it short for now) and press RETURN. Press RETURN again to skip a line between the title and the paragraph.

Now we must move around again, as we wish to return to the end of the paragraph. Press ESC and see how we have gone into EDITING MODE again. Instead of using arrow keys to move around this time, we'll use a shortcut. Simply press E (for end) and

BEST COPY AVAILABLE

the cursor goes to the end of the text. Try pressing B (for beginning) and see what happens. It is best to use these keys to move around when the text is lengthy. Other quick-move keys will be explored later.

Now let's add one more paragraph to the text.

Don't forget to skip a line after the first paragraph by pressing RETURN twice. Make sure you write enough so that the two paragraphs cause the writing to scroll upward. Press ESC to enter the edit mode when you are through.

Two more keys which allow you to move around quickly in edit mode are U (for up) and D (for down). Try these and see what happens. These keys allow you to move up and down 12 screen lines at a time.

Now for the exciting part. We shall save the text that was entered and go make a paper copy (also known as "hard copy"). You are in edit mode.

Examine the menu at the top of the screen. Notice that the two apples are pictured in the highlighted box.

Use the full apple to move the highlighting from the ERASE menu choice over to the right. Keep moving it until TRANSFER MENU is highlighted. If you go past it, use the other apple key to go back.

When TRANSFER MENU is highlighted, press the RETURN key.

Once that is done, we can go on to the next step. We wish to SAVE our file, so press the white apple key until the SAVE selection is highlighted, then press RETURN. The question "DO YOU WANT A CATALOG" will appear.

Answer the question by typing N, for "no." If you want to see the names of all the files that are stored on your disk, you would press Y, for "yes." Press RETURN.

All you need to do now is follow the instructions at the top of the screen. Make up a name for the file, and a password if you wish, BUT DON'T FORGET THE PASSWORD IF YOU DO!

IMPORTANT: When you save a file, you simply put a copy of your file onto the disk. The original information in the file *is still in the computer's memory.*

If you want to get a copy of your file on paper (called "hard copy"), you must use one of the PRINT options from the onscreen menu. PRINT-DRAFT will give you a "draft" copy which will be in forty columns and triple spaced automatically. PRINT-FINAL will allow you to have some control over spacing, pagination and other features.

IMPORTANT NOTE: If you are not working on a computer that is attached to a printer (the professional word to use is "interfaced"), you must remove your disk and walk over to a computer that has a printer hooked up to it. Again, boot up the Bank Street Writer disk and then insert your own disk. Press ESC to enter the edit mode, and highlight the TRANSFER MENU to enter transfer mode. Then select the RETRIEVE option and press RETURN. Enter the name of your file and follow screen directions. ***You must use the RETRIEVE option anytime that you are starting on a new computer and need to "load" one of your files into the computer's memory.***

Once you have your file in memory (it will still be there if you have not changed computers), and have selected PRINT-FINAL, the next task is to answer a lot of questions, some of which have little meaning to inexperienced users.

Notice that each question is followed by a likely answer. This is called a DEFAULT VALUE. It means that if you don't know the answer to a particular question or if the default is the answer that you want to give, you can just press RETURN and the question will be answered for you with a standard value.

For example, the second question deals with spacing. If you just press RETURN, you will get a single-spaced document. If you type a 2 in response to this question, it will be double-spaced instead.

Continue to answer questions this way. You may answer "no" when asked if the pages should be numbered. A short letter or story would look silly with a number 1 at the bottom of the page.

When you reach the last question, asking you to prepare the printer, do the following:

1. Be sure the printer is OFF. (If you try to move the roller when the printer is ON, you will notice some resistance. It is hard on the printer if you move the roller against the resistance.) Turn the carriage in a forward direction by hand until the perforated tear line between the two sheets is lined up with the printer head.
2. Turn the printer ON.
3. Press the RETURN key as asked for by the Command line on the screen. The printer will then print your file.
4. When your file has been printed, tear off the printed pages and remove the lacy edges by carefully folding them and then tearing gently.

BEST COPY AVAILABLE

There are many options to be explored on *Bank Street Writer*. What is important is that you now know enough to operate it by yourself with the help of the manual. To explore some of these capabilities, flip the disk over (yes, this one is two-sided!) and go through the steps of the *Bank Street Writer TUTORIAL*. It will help you see what some of these other menu options do.

As you practice using the word processor, you will get used to its capabilities and limitations. You should feel encouraged to type some of your college papers on it. This is excellent practice, and is important because you are doing *relevant* work on it. Keep this in mind when you teach other people (especially kids) to use a computer!

Self-Quiz #2

This is another short self quiz to help you review some of the new vocabulary that you have encountered in the course.

1. CAI stands for
    - a) computers and instruction.
    - b) cultural aids to intelligence.
    - c) computer-assisted instruction.
    - d) complex algorithmic illustration.

2. CAI programs which teach new skills are generally called
    - a) drill-and-practice.
    - b) utilities.
    - c) tutorials.
    - d) simulations.

3. To view the program steps which are stored in the memory of the computer, the user should type
    - a) RUN.
    - b) LIST.
    - c) LOAD.
    - d) CATALOG.

4. In order to store the program named DEMO on a disk, the user should type
    - a) SAVE DEMO.
    - b) LIST DEMO.
    - c) RUN DEMO.
    - d) DEMO SAVE.

5. The following is a type of utility program:
    - a) a spelling drill-and-practice program.
    - b) a science simulation of atoms and molecules.
    - c) a word processing program.
    - d) a space invaders game.

6. Programs which are saved on a disk
    - a) are erased when they are loaded into memory.
    - b) can only be used once.
    - c) can be used only on one computer.
    - d) can be used over and over again.

BEST COPY AVAILABLE

7. Teachers seldom write elaborate, complex programs because
    a) they cannot learn programming.
    b) they do not have much time.
    c) they do not understand the educational implications.
    d) they do not believe that computers can aid the instructional process.

8. Computers CANNOT be used to
    a) help keep lengthy records.
    b) give students practice with old skills.
    c) illustrate concepts graphically.
    d) give a total educational program.

9. Many students think that computers are patient and friendly because
    a) they produce nice color pictures.
    b) they do not show anger when asked to repeat a question or when the student takes a long time answering.
    c) they always use the student's name.
    d) most programs use a game approach.

10. Early CAI programs were math drill-and-practice because
    a) they were easiest to program.
    b) most computing teachers were math teachers.
    c) computers lend themselves best to mathematical uses.
    d) the greatest need was for remedial mathematics.

Answers to Self-Quiz #2

1. *c* - While all the other answers *could* be true, the correct interpretation of the acronym CAI is Computer Assisted Instruction.

2. *c* - A "tutor" is one who teaches. New knowledge usually comes from a *tutorial.*

3. *b* - You tell the computer to LIST the program statements stored in its memory.

4. *a* - To store a program, you must instruct the computer to SAVE it.

5. *c* - Word processing serves as a *utility* to help a teacher (or any other writer).

6. *d* - Just like a tape recording, the program is stored until we choose to erase it, so it can be used over and over again.

7. *b* - Sophisticated programming is a highly skilled and time-consuming process. Most teachers simply don't have the time to give to elaborate programming.

8. *d* - Education consists of much more than learning facts and practicing skills. It is unrealistic to expect a computer to carry out all the varied tasks involved in a total education program.

9. *b* - Just like all machines, a computer does not have emotions and is therfore capable of virtually infinite patience. Of course, the program must be skillfully written if it is to repeat questions and wait for input at the appropriate places.

10. *a* - The structure of simple "drill-and-practice" allows it to be programmed reasonably easily. This does not mean, however, that all drill-and-practice programs are unsophisticated.

BEST COPY AVAILABLE

Readings

Computer Literacy: Talking and Doing

(Editor's Message)

Dave Moursund

The U.S. Office of Education has funded a national project to define computer literacy and to develop an instrument that could be used to measure the computer literacy of students, teachers, principals and superintendents. The Educational Testing Service of Princeton, New Jersey is the prime contractor on this project and the Human Resources Research Organization of Alexandria, Virginia is a subcontractor for a major piece of the work.

In January 1983 a distinguished group of "national experts" met in Washington, D.C. to express their opinions on ideas going into the development of the definition and measurement instruments. I found the meeting both enjoyable and frustrating.

The essence of the problem is defining computer literacy. The group of nine national experts seemed to divide into two camps. One camp essentially thinks of computer literacy as a talking-level of knowledge. This includes knowing some computer history; knowing definitions of computer-related words; knowing some applications of computers in business, government and industry; knowing about social and ethical issues; and so on. Many of these objectives are stated, "The student is aware of . . ." Of course this talking-level-of-knowledge camp wants students to have some hands-on computer experience and hence a modest level of skill in actually using a computer.

The second group admits that the ideas of the first group are important, but insists that the most important aspect of computer literacy is being able to make a computer *do* things. This group then is further divided as to what this might mean. Does it mean being able to program in Logo, BASIC or Pascal, or does it mean knowing how to use a word processor, an information retrieval system and an electronic spreadsheet?

Those of you who know me personally or who have followed my writings are aware that I support both groups, but most strongly support the "doing" parts of a definition of computer literacy. Within that group I am moderately supportive of the "traditional language computer programming" subgroup and very strongly supportive of students learning to use applications packages.

I believe that we are at a critical junction. If the "talking-knowledge" definition wins out, it will be a substantial setback to the progress of computers in education. Imagine if a talking-knowledge group won out in reading and writing literacy or in mathematics literacy. Students would not have to be able to read, write or do mathematics. Rather, we would assess their literacy (presumably via oral testing) by determining if they knew some applications of reading, writing and arithmetic. We would determine if they had some historical knowledge of these areas, had seen and touched a book, had been read to from a book, and so on. I suppose that the ethics of book stealing would be discussed, as would be the threat to privacy of actually using books to store information about people. Numerology would perhaps be a required course and students could make oral presentations relating numerology to astrology.

All of this is silliness, of course, as is limiting computer literacy to a talking level of knowledge. I believe the major issue will be the question of programming versus using applications packages within the "doing" group. I am quite experienced in arguing that all students should receive instruction in computer programming. Such instruction is an excellent vehicle for learning and practicing important ideas such as problem solving, top-down problem analysis, representation of algorithms, algorithmic thinking, repetition, modularity, bugs and debugging, documentation. My fear is that the typical teacher of computer programming may treat these topics as incidentals and produce "hackers" instead of computer literates.

At the current time I support the teaching of computer programming at any grade level, provided the teacher is competent in the computer field—that is, has a good knowledge of all of the ideas listed in the previous paragraph and integrates them into the instruction; that the instructional materials have these same characteristics; and that the totality of hardware and software available are adequate to support this type of instruction.

Unfortunately, this seldom occurs. The basis for my strong support of students learning to use applications packages is that less teacher knowledge and less high-quality instructional materials are needed. Moreover, this is clearly the sense of direction of the computer field. The computer-using public of the year 1990 or 2000 will not know a programming language such as Logo, BASIC or Pascal. Their ability to use computers will not depend upon knowing how to program in these or other high-level programming languages.

What *will* be important is knowing how to utilize the capabilities of a computer and integrating this knowledge into one's overall knowledge and performance. The student who knows how to write will know how to use a word processor as an aid to writing. The student who knows how to read will know how to use a computerized information retrieval system to obtain materials to be read. The student who knows the meaning and use of equations or graphs will know how to use a computer to solve equations and draw graphs. The student interested in composing music will know how to use a computer-assisted music composition and performance system. All students will learn to use electronic spreadsheets and other general-purpose aids to problem solving.

I believe that this is the key to computer literacy. It is a "doing" level of knowledge, with this knowledge integrated into the totality of a student's knowledge and performance. This type of computer literacy can be taught now and can be measured. I believe that it is essential that those who would define and measure computer literacy see that this is where we are headed and that this must be a significant part of any modern definition of computer literacy.

END■

January 1983

I Don't Know

Dave Moursund

A few weeks ago I was walking down the display aisle at a large computers-in-education conference. Within a period of a few minutes I was "accosted" by three different individuals.

> Individual #1: "What do you think of—(a particular piece of courseware)? Don't you think it's really neat?"
>
> My response: "I'm sorry, I haven't tried it yet."
>
> Individual #2: "What do you think of—(a particular new piece of hardware)? Do you feel it will really take hold?"
>
> My response: "I'm sorry, I haven't seen it yet."
>
> Individual #3: "My school wants to use computers for—(briefly describes a particular problem). What do you recommend?"
>
> My response: "I'm sorry, I don't know very much about specific hardware and software in that particular area."

Do you see the pattern? It is one in which I am continually apologizing for my lack of expertise; I am continually made to feel inadequate.

It isn't as though I don't know quite a bit about the field of computers in education. I taught my first course to high school students in this field in 1963, and I began to teach teachers in this field in the summer of 1965. Since then I have worked very hard to learn more about computers, computers in education and teacher education. I am a good student and I have made substantial progress.

Part of the problem is psychological, and it is here that my past comes back to haunt me. My bachelor's, master's and doctor's degrees are all in mathematics, as were my initial teaching experiences. I was taught (that is, strongly indoctrinated) that a bachelor's degree in mathematics qualifies a person to teach precollege mathematics. If one is an especially strong student, the bachelor's degree earns one a graduate teaching assistantship and minimally qualifies one to teach freshman college algebra courses.

With a master's degree and more teaching experience, the graduate teaching assistant is allowed to teach an introductory calculus course. The "really advanced" graduate student in mathematics might even be allowed to teach sophomore level courses!

The general model that I learned for math teaching was that one should be totally competent. A good math teacher must be able to explain every topic and solve every problem arising in a course. Indeed, the goal was to be able to do this in real time, off the top of one's head. It was not appropriate to respond to a question with, "I don't know."

I don't know—let's learn together! It seems like a simple idea, and it is a healthy attitude for all educators. Roughly speaking, I find that elementary teachers most easily implement the "I don't know—let's learn together" idea, followed by junior high school teachers, high school teachers and college teachers, with university teachers at the bottom of the list. I can see why I have trouble—especially when my math background is factored in!

"New" math notwithstanding, the content of the K-16 math curriculum is quite old. It is difficult to find a major topic in this curriculum that is less than 50 years old, and most of the content is hundreds of years old.

Computers in education is just the opposite. Almost everything is less than 50 years old and much of it is less than a couple of years old. Hundreds of companies are developing new hardware and software. Literally thousands of people are trying out new ideas, developing new curricula and carrying on research. Computers are both a relevant aid to problem solving and a source of problems in every academic discipline at every academic level. This relevance is growing rapidly.

So, what is a middle-aged, retreaded mathematician to do? Certainly I can continue to learn. That is essential.

But it is equally essential to overcome the psychological barrier. I need to learn to say, "I don't know—let's learn together," **without following it with an apology.** There is nothing wrong with not knowing about the latest piece of hardware or software. There is nothing wrong with not carrying around in one's head all of the fine details of a particular machine that some bright 14-year-olds seem to memorize so easily. I am still a competent person even if a 14-year-old knows many things about computers that I don't know.

For me, what is especially important is learning to take advantage of my strengths, such as wisdom and experience gained through the past 46 years. I have a greater breadth and depth of knowledge than most of my students. I have had a greater range of experiences, and I have had more practice in learning to learn. If I keep these facts firmly in mind, I can more easily say, "I don't know—let's learn together." For me, that is an important goal.

END■

BEST COPY AVAILABLE

Informational Technology and Its Impact on American Education

(Summary Report)

Office of Technology Assessment
Congressional Board of the 97th Congress
TED STEVENS, Alaska, Chairman
MORRIS K. UDALL, Arizona, Vice Chairman

Senate

ORRIN G. HATCH
Utah

CHARLES McC. MATHIAS, JR.
Maryland

EDWARD M. KENNEDY
Massachusetts

ERNEST F. HOLLINGS
South Carolina

HOWARD W. CANNON
Nevada

House

GEORGE E. BROWN, JR.
California

JOHN D. DINGELL
Michigan

LARRY WINN, JR.
Kansas

CLARENCE E. MILLER
Ohio

COOPER EVANS
Iowa

JOHN H. GIBBONS
(Nonvoting)

Advisory Council

CHARLES N. KIMBALL, Chairman
Midwest Research Institute

EARL BEISTLINE
University of Alaska

CHARLES A. BOWSHER
General Accounting Office

CLAIRE T. DEDRICK
California Land Commission

JAMES C. FLETCHER
University of Pittsburgh

S. DAVID FREEMAN
Tennessee Valley Authority

GILBERT GUDE
Library of Congress

CARL N. HODGES
University of Arizona

RACHEL McCULLOCH
University of Wisconsin

WILLIAM J. PERRY
Hambrecht & Quist

DAVID S. POTTER
General Motors Corp.

LEWIS THOMAS
Memorial Sloan-Kettering
Cancer Center

Director
JOHN H. GIBBONS

The Technology Assessment Board approves the release of this report. The views expressed in this report are not necessarily those of the Board, OTA Advisory Council or of individual members thereof.

Foreword

Over the last decade, American education has come to face a number of new demands that must be met with limited resources. Many of these new demands arise from the growing dependence of our society on high technology as a basis for domestic economic growth, international competitiveness and national security. In October 1980, the House Committee on Education and Labor, its Subcommittee on Special Education, and the Subcommittee on Science, Research and Technology of the House Committee on Science and Technology asked OTA to examine the extent to which information technology could serve American needs for education and training.

This report documents two basic sets of conclusions:

1. The so-called *information revolution*, driven by rapid advances in communication and computer technology, is profoundly affecting American education. It is changing the nature of what needs to be learned, who needs to learn it, who will provide it, and how it will be provided and paid for.

2. Information technology can potentially improve and enrich the educational services that traditional educational institutions provide, distribute education and training into new environments such as the home and office, reach new clients such as handicapped or homebound persons, and teach job-related skills in the use of technology.

The OTA report provides an overview of the issues relating to the educational applications of the new information technologies. It examines both the demands that the information revolution will make on education and the opportunities afforded by the new information technologies to meet those demands. Rather than focusing on a single technology, it examines the full range of new information products and services such as those based on the combined capabilities of computers, telecommunications systems and video technologies. Similarly, the report surveys a broad range of educational providers, and examines how the application of information technologies may affect their abilities to provide education and their respective educational roles.

OTA acknowledges with thanks and appreciation the advice and counsel of the panel members, contractors, other agencies of Government, and individual participants who helped bring the study to completion.

JOHN H. GIBBONS
Director

Informational Technology and Its Impact on American Education Advisory Panel

Willis Adcock
  Assistant Vice President
  Texas Instruments Inc.

Joel N. Bloom
  Director, Franklin Institute
  Science Museum and Planetarium

Colleen Cayton
  Maxima Corp.

Robert L. Chartrand
  Congressional Research Service

Mark Curtis
  President, Association of American Colleges

L. Linton Deck
  Fairfax County Schools

Sam Gibbon
  Bank Street College

Harold Howe, II
  Harvard Graduate School of Education

Robert Hoye
  Director, Instructional Technology
  University of Louisville

Judith Lozano
  Superintendent of Southside School District
  San Antonio, Texas

Maurice Mitchell
  Chairman of the Board
  National Public Radio

Sarah Resnick
  President, Media Systems Corp.

Vic Walling
  Stanford Research Institute

Nellouise Watkins
  Director, Computer Center
  Bennett College

Joe Wyatt
  Vice President for Administration
  Harvard University

Acknowledgments

The following individuals contributed as contractors or reviewers during the course of this study

Richard Ballard, TALMIS Corp.
Charles Benton, Films, Inc.
George Blank, Creative Computing
Robert C. Bowen, McGraw-Hill Book Co.
Dee Brock, Public Broadcasting Service
John Cameron, Department of Commerce
Sylvia Charp, School District of Philadelphia
Richard Diem, University of Texas at San Antonio
Seymour Eskow, Rockland Community College
Albert Goldberg, WCISD
Jim Johnson, University of Iowa
Valerie Klansek, Upper Midwest Region
  Resource Center
Paul Larkin, Prince Georges Community College
Joe Lipson, WICAT
Tom Loftis, Office of Personnel Management
Arthur Melmed, Department of Education
Andy Molnar, National Science Foundation

B. Otte, National Institute of Education
Robinson, Scholastic, Inc.
canland, U.S. Navy
Patsy Vyner, Close Up Foundation
Fred Wood, Department of Agriculture
Andy Zucker, Department of Education

Informational Technology and Its Impact on American Education Project Staff

John Andelin, Assistant Director, OTA
Science, Information and Natural Resources Division

Stephen E. Doyle* and Sam Hale,** Interim Program Manager
Communication and Information Technologies Program

Fred W. Weingarten, Project Director

Prudence S. Adler, Assistant Project Director***

Dorothy Linda Garcia, Analyst

Beth A. Brown, In-House Consultant

Susan F. Cohen, Congressional Fellow

Linda G. Roberts, Consultant (Senior Associate,
Department of Education, on detail)

Elizabeth Emanuel, Administrative Assistant

Shirley Gayheart, Secretary

Jeanette Contee, Wordprocessor

Contractors

Christopher Dede, University of Houston

Beverly Hunter, Brian K. Waters and Janice H. Laurence,
Human Resources Research Organization

Sharon Lansing, Consultant

Kathryn M. White, Editor, Writer

Renee G. Ford, Tifford Producers, Ltd., Editor, Writer

Deeana Nash, Collingwood Associates

OTA Publishing Staff

John C. Holmes, Publishing Officer

John Bergling Kathie S. Boss Debra M. Datcher Joe Henson

---

m Manager served through February 1981
m Program Manager from March 1981 through March 1982. Fred W. Weingarten has served as Program Manager since then.
ed as Assistant Project Director since March 1982.

BEST COPY AVAILABL

Modern society is undergoing profound technological and social changes brought about by what has been called the information revolution. This revolution is characterized by explosive developments in electronic *information technologies* and by their integration into complex information systems that span the globe. The impacts of this revolution affect individuals, institutions and governments—altering what they do, how they do it, and how they relate to one another.

If individuals are to thrive economically and socially in a world that will be shaped, to a large degree, by these technological developments, they must adapt through education and training. Already there is evidence of demands for new types of education and training, and of new institutions emerging to fill these demands. The historical relationship between education and Government will be affected by the role that Government plays in enabling educational institutions to respond to the changes created by these technologies.

Background

Historically, the Federal Government's interest in educational technology has been sporadic—rising as some promising new technology appeared and falling as that technology failed to achieve its promise. Attention was focused, moreover, on the technology itself and not on the broader educational environment in which it was to be used. In the late 1960s, for example, the Federal Government funded a number of research and development projects in the use of computer-assisted instruction (CAI). Interest in the projects waned, however, given the high costs of hardware and curricula and the failure to integrate computer-based teaching methods into the institutional structure of the school.

Computer-based education is the use of computers for educational purposes. It includes:

1. *Computer-Managed Instruction (CMI).*—Wherein learning takes place away from the computer, while the computer scores tests, interprets results, advises the student what to do next, and manages student records and other information.
2. *Computer-Assisted Instruction (CAI).*—Wherein the student receives individualized instruction by interacting via a computer terminal with the instructional material logic stored in a computer.

Over the last decade, Federal funding for research and development (R&D) in educational information technology has dropped precipitously. At the same time, development and applications of information technology have advanced rapidly in many sectors. Public schools, beset by problems that such technology might mitigate, have lagged behind in adapting to technological changes. In view of this situation, OTA was asked in October 1980 to reexamine the potential role of new information technology in education. The assessment was initiated at the request of: 1) the Subcommittee on Select Education of the House Committee on Education and Labor; and 2) the House Subcommittee on Science, Research, and Technology of the Committee on Science and Technology.

YOUR FIRST COMPUTER

Photo credit. © Ted Spiegel, 1982.

This report examines both the demands the information revolution will make on education and the opportunities afforded to respond to those demands. Included in its scope are a survey of the major providers of education and training, both traditional and new, and an examination of their changing roles. The full range of new information products and services rather than any single technology is examined, since the major impact on education will most likely stem from the integration of these technologies into instructional systems.

For this report OTA has defined *education* to include programs provided through a variety of institutions and in a variety of settings, including public schools; private, nonprofit institutions that operate on the elementary, secondary and postsecondary levels; proprietary schools; training and education by industry and labor unions; instruction through the military; and services provided through libraries and museums or delivered directly to the home. *Information technology* is defined to include communication systems such as direct broadcast satellite, two-way interactive cable, low-power broadcasting, computers (including personal computers and the new hand-held computers), and television (including video disk and video tape cassette).

BEST COPY AVAILABLE

The assessment was premised on three initial observations and assumptions:

- The United States is undergoing an information revolution, as documented in an OTA assessment, *Computer-Based National Information Systems.*
- There is a public perception that the public schools are "in trouble," and are not responding well to the normal educational demands being placed on them. Public schools in many parts of the country are faced with severe economic problems in the form of rapidly rising costs and reduced taxpayer support. These pressures are forcing a new search for ways to improve the productivity and effectiveness of schooling.
- A host of new information technology products and services that appeared capable of fulfilling the educational promises anticipated earlier are entering the marketplace with affordably low cost and easy accessibility.

Findings

OTA found that the real situation is far more complex than assumed above. In summary, the assessment's findings are:

- The growing use of information technology throughout society is creating major new demands for education and training in the United States and is increasing the potential economic and social penalty for not responding to those demands.
- The information revolution is creating new stresses on many societal institutions, particularly those such as public schools and libraries that traditionally have borne the major responsibility for providing education and other public information services.
- Information technology is already beginning to play an important role in providing education and training in some sectors.
- Information technology holds significant promise as a mechanism for responding to the education and training needs of society, and it will likely become a major vehicle for doing so in the next few decades.
- Much remains to be learned about the educational and psychological effects of technological approaches to instruction. Not enough experience has been gained with the new information technology to determine completely how that technology can most benefit learners or to predict possible negative effects of its use. Given this insufficient experience, caution should be exercised in undertaking any major national effort, whether federally inspired or not, to introduce these new technologies into education.

The Information Society

Role of Information

For the foreseeable future, information technology will continue to undergo revolutionary changes. The microprocessor—an inexpensive, mass-produced computer on a chip—will become ubiquitous in the home and office—not only in the easily identifiable form of the personal computer or word processor, but also as a component of numerous other products, from automobiles to washing machines and thermostats. High-speed, low-cost communication links will be available in such forms as two-way interactive cable, direct broadcast from satellites and computer-enhanced telephone networks. New video technologies such as video disks and high-resolution television will be available. These technologies will be integrated to form new and unexpected types of information products and services, such as videotex and on-line information retrieval systems that can be provided over telephone or air waves directly to the home.

Photo credit © Ted Spiegel, 1982

Personal-type computers are used for instruction in many classrooms throughout the nation.

BEST COPY AVAILABLE

It is impossible to predict which of these technologies and services will succeed in the competition for consumer dollars, or which will appeal to particular markets. It is, however, reasonable to conclude that they will radically affect many aspects of the way society generates, obtains, uses and disseminates information in work and leisure.

The growing importance of information itself drives and is driven by these rapid technological changes. Until a few decades ago, the information industry—that industry directly involved with producing and selling information and information technology—was relatively small in economic terms. It is now becoming a major component of the U.S. economy. While most economists still talk about the traditional economic sectors—extractive, manufacturing and service—some now have begun to define and explore a fourth, the information sector. One analysis has shown that this new sector, if defined broadly, already accounts for over 60 percent of the economic activity of the United States.

Many firms involved directly with information are large and growing. Two of the largest corporations in the world, AT&T and IBM, principally manufacture information products and provide information services. Moreover, business in general is beginning to treat information as a factor of production that takes its place beside the conventional factors of land, labor and capital. In addition, the Government is beginning to treat information as an important element of national security. While defense officials have always been concerned about the disclosure of military information—such as troop movements or weapons design—they are now also concerned about the international leakage of more general U.S. scientific and technical information that other countries could conceivably use to pursue economic or military goals that are in contrast to our own.

In addition to serving as an economic good, access to information is becoming increasingly important for individuals to function in society effectively as citizens, consumers and participants in political processes. Relations with government at all levels are becoming more complex—whether they involve dealing with the Internal Revenue Service, applying for social benefits and services, or seeking protection from real or perceived bureaucratic abuse. Individuals are confronted with the need to evaluate more sophisticated choices and to understand their rights and responsibilities under the laws and regulations intended to protect them in the marketplace.

Information Technologies

The rapid evolution of the following technologies in the last few decades has shaped the information revolution:

Cable

Cable systems—wherein data and programs are transmitted over a wire rather than through airwaves—are growing rapidly. The newer systems offer more channels, and some offer two-way communication.

Satellite Communication

Satellites have stimulated development of new types of television networks to serve cable subscribers and earth station owners with specialized programming.

Digital Telephone Network

The shift to digital transmission will allow telephone lines to carry more information at higher speed and with greater accuracy, providing better linkage of information between computer terminals.

Broadcast Technologies

Some distribution technologies in the entertainment market may also have important potential educational uses. For one, the *direct broadcast satellite* can transmit a program directly to a home or office, bypassing a cable system. For another, *low-power stations*, which restrict transmission to a limited geographical range, provide a low entry cost to licensees and are subject to less regulation than are traditional broadcast stations.

Computers

The design and uses of computers have advanced to the point where there is now a mass consumer market for computers and computer software. Moreover, networks that link privately owned computers have expanded access to information. *Desktop computers* are becoming more common in the home, the small business, and formal educational settings. The use of *hand-held computers*, cheaper and more portable than desktop computers, has also increased. Along with computer development have come advances in the interface between humans and computers—input/output technology. Input technology is the process of putting information into the computer—either by typing it, speaking to the computer or showing the computer pictures. Developments in output technology, or "peripherals," are occurring in the areas of low-cost printers, graphics (particularly color graphics), and voice.

Storage Technology

Data programs are stored on a variety of media for use in the computer: silicon chips, floppy disks and hard disks. Improvements are being made in such technology for both large and small computers.

Video Technology

Significant developments in several areas of video technology are likely in this decade. *Video cassette recorders* are already important consumer devices. The *filmless camera*, which combines video and computer technology to "write" a picture on a very small, reusable floppy disk, may soon be available.

Video Disks

Resembling a phonograph record, a disk that stores television programming is of considerable interest to educators. It is durable, inexpensive to produce, and capable of storing a large amount of data and programs.

Information Services

Several of the aforementioned information technologies are now being integrated into information systems. For example, several countries now use the existing television broadcast medium to bring information services to homes and offices. Using a *teletext* system, the user can select a page for special viewing as it is transmitted in segments over the air. In a videotex system the user can preselect a page from the central system for immediate viewing. Closely related to videotex are the *information networks* that provide owners of desktop computers and terminals with access to computer and data services and to one another over communication networks. Through *electronic conferencing*, geographically separated individuals can participate in meetings. Variations include *audio conferencing*, which uses telephone lines; *video conferencing*, which supplements the voice connection with television images; and *computer conferencing*, which involves transmitting messages through a central computer that then distributes them as requested.

Impacts on Institutions

Impacts from the information revolution are being felt by government at all levels and by the military, industry, labor unions and non-profit service institutions. Traditional services provided by these institutions now overlap in new ways and offer a wide variety of new services based on information technology. For example, firms as diverse as investment houses and retail stores now compete with banks by providing a variety of financial services. Banks, on the other hand, are beginning to compete with computer service bureaus in providing more general on-line information services to businesses and homes.

The U.S. Postal Service, along with Congress and a variety of Federal executive and regulatory agencies, is considering the degree to which it should compete with private telecommunications firms in the provision of electronic mail services. Large computer firms such as IBM are moving toward direct competition with traditional telecommunication common carriers such as AT&T for the provision of information services. Telephone companies may offer "electronic yellow pages" that could rival the classified advertising business of newspapers.

Those institutions principally concerned with the collection, storage or transfer of information will feel the greatest effects. They include both private sector firms —in fields such as publishing, entertainment and communications—and public or nonprofit organizations such as libraries, museums and schools. How they handle their product—information—may differ from the handling of tangible goods by other institutions because information has characteristics that differentiate it from tangible goods. For example, information can be reproduced easily and relatively inexpensively. It can be transported instantly worldwide and presumably can be transferred without affecting its original ownership. Thus, copyright or other forms of protection for intellectual property—data bases, programs or chip designs—is important to the growth of the information industry.

While the business of selling information has always existed in some form—e.g., book publishing, newspapers or broadcasting—the growth of this sector and its movement into electronic forms of publishing will create conflicts with traditional societal attitudes about information. The concept of information as a public good whose free exchange is basic to the functioning of society is inherent in the first amendment to the Constitution and underlies the establishment of public libraries and schools. This concept conflicts with the market view of information, which recognizes that there are inherent costs in the provision of information. Adopting new information technologies will entail extra costs that must be borne somehow by the users of those technologies.

The conflict between the view of information as a market good and the view of it as a "public good" affects public institutions in a number of ways. Public nonprofit institutions find themselves increasingly in competition with private profitmaking firms that offer the same or similar services. Institutions such as libraries, schools and museums are beginning to feel pressure to incorporate both nonprofit and income-generating offerings in their own mix of services. To the extent that previously free or very low-cost and widely available information services such as education move into the private marketplace, access to them may become limited, either because of their cost or because of their restricted technological availability. Periodicals previously available at newsstands, for example, may be available in the future only via computer or video disk.

New Needs for Education and Training

The information revolution places new demands on individuals, changing what they must know and what skills they must have to participate fully in modern society. It may also be increasing the social and economic prices that will be paid by those who do not adapt to technological changes. For instance, spurred by increasing domestic and international economic competition, U.S. industry is expected to adopt computer-based automation in a major way. Computer-aided design, robotics and other new computer-based manufacturing technologies will, within the next decade, transform the way goods are manufactured. Automation will not be restricted to the factory, however. Office automation will, according to some, have an even more revolutionary effect on management and on clerical work in business. Over the longer term, even the service professions, such as law and medicine, will be transformed.

While some sociologists suggest that the effect will be to "deskill" labor by lowering the skill requirements for workers, more anticipate that a greater premium will be placed on literacy, particularly technological and information literacy. The latter argue that an increasing number of jobs will be in the information sector or will require the use of information systems. Moreover, new forms of production and information handling will create new jobs requiring new skills. Vocational education and industrial training programs will be needed to teach the skills for jobs such as robot maintenance or word processing.

An advanced information society will place a premium on skills oriented toward the creation of new knowledge and the design of new technologies. Thus, while there is some current debate about a possible surplus of college graduates, generally speaking many experts see a growing gap between the demand and supply of graduates in engineering and science, and particularly in computer engineering and science.

A key element in all of these educational needs is that they will constantly change. In a rapidly advancing technological society, it is unlikely that the skills and information base needed for initial employment will be those needed for the same job a few years later. *Lifelong retraining is expected to become the norm for many people.*

Case Studies on Information Technology

In addition to using existing information for this assessment, OTA undertook case studies designed to gain insights into the successful application of information technology in education. Accordingly, OTA examined well-established programs in public school systems, industries, libraries, museums, the military, special education and direct to the home markets nationwide. These case studies are presented in the appendix of the full report. Many of the findings presented in this assessment reflect observations made in these studies. *The most important of these observations is that information technologies can be most effectively applied to educational tasks when they are well integrated in their institutional environments.*

Potential Technological Solutions

OTA found little evidence of current hardware limitations that would limit the applicability of technology to education and, hence, call for major research efforts. Continuing research in the general fields of computer science and engineering, coupled with innovative private sector development, will provide the necessary hardware base. The only exception is the area of technology for the handicapped, where it is not clear that the opportunities for developing specialized technology could be met without some Federal support for R&D. There does appear to be a need, however, for R&D focused on developing new techniques and tools for software development, human/machine interface, and improving the understanding of cognitive learning processes.

If properly employed, information technology has certain characteristics that suggest it will be invaluable for education. For one, information technology may be the only feasible way to supplement teaching capability in schools faced with reduced teaching staffs and larger class sizes. For another, information technology is capable of distributing education and training, both geographically and over time. Services can be provided in the home, at work, in a hospital, or in any other location where and when they may be needed.

Many of the electronic media, such as video disks or microcomputers, allow learners to use them at their convenience, instead of being locked into specifically scheduled times. Computer-based analysis, combined with a flexible, adaptive instructional system could diagnose and immediately respond to differences in learning strategies among students and, hence, could be more educationally effective. Finally, much work has been done on using information technology to improve the ability of foreign students and the physically and mentally handicapped to communicate.

Some experts suggest that the use of computers by students teaches them new ways of thinking and new ways of solving problems that may be more appropriate in an information age. They suggest that a generation that grows up with computers will have a significant intellectual advantage over one that does not.

Photo credit © Ted Spiegel, 1982

Many educators criticize such a view as being too technology-centered. At the very least one can predict, however, that computer and computer-based information services will be ubiquitous by the next century, and that learning how to use them effectively is a basic skill that will be required for many and perhaps most jobs. (In response to this view of future skill requirements, many schools have placed a high priority on computer literacy as the first instructional use of the computer.)

Although experience with educational technologies has demonstrated that they offer a variety of potential benefits, it has also demonstrated that technology cannot, by itself, provide solutions to all educational problems, nor should it be imposed on an educational system without sensitivity to institutional and societal barriers that could prevent the realization of educational benefits. These barriers include:

Institutional Barriers

New educational technology must be designed for ease of integration into the schools and other educational institutions that will use it. Some adaptations of curricula, schedules and classroom organization will be needed, but the changes are not likely to be extreme.

Teacher Training

Widespread use of technology in the classroom will require that teachers be trained both in its use and in the production of good curriculum materials. Too few teachers are so qualified today. Schools maintain that they are already faced with a shortage of qualified science and mathematics teachers (those most likely to lead the way in computer-based education). Furthermore, there is little evidence that most of the teacher training colleges in the United States are providing adequate instruction to new teachers in the use of information technology.

Lack of Adequate Software

OTA found general widespread agreement that, with few exceptions, the quality of educational software—curriculum material designed for educational technology—now available was, in general, not very good. Curriculum providers do not yet use the new media to full advantage for several reasons. In the first place, many of the technologies are still new. It takes time to learn how to use them, and the early attempts suffer from this learning process. Second, production of high-quality educational software is expensive. Some large firms that have the necessary capital to produce educational software hesitate to risk developmental money in a relatively new and uncertain market.

Third, the programmers and curriculum experts qualified to produce educational software are in short supply. Finally, some firms cite the lack of adequate property protection—e.g., copyright, patents—for their information products as a barrier to investment in development.

Skepticism About Long-Term Effects

Some educators are seriously concerned that the long-term effects on learning of substituting technology for traditional teaching methods are not sufficiently understood. While acknowledging that computers or other technologies may have some limited utility in the classroom for drill and practice, or for instruction in computer literacy, they fear that any widespread adoption of technology for education could have deleterious effects on the overall quality of learning.

Cost

Even though the cost of computer hardware and communication services is dropping, investment in educational technology still represents a substantial commitment by financially pressed schools. Costs of software are likely to remain high until a large market develops over which providers can write off developmental costs. In some cases the cost of information products and services may be passed on to users for the first time.

Policy Issues and Options

Issues

The impact of information technology on education will confront Congress with a number of important policy decisions in several areas:

- *Education and training for economic growth:* OTA found that trends in automation and the growth of the information sector of the economy will probably present the United States with severe manpower training problems over the next decade. These will include a persistent shortage of highly trained computer scientists, engineers and other specialists; a need for retraining workers displaced by factory and office automation; and a need for a more technologically literate work force. Congress must decide what Federal response to these national needs would be both appropriate and effective.
- *Redressing inequities:* In both the OTA study on national information systems and in this assessment, OTA found concern that a significant social, economic and political gap could develop between those who do and those who do not have access to, and the ability to use, information systems. People who cannot make effective use of information technology may find themselves unable to deal effectively with their government and to obtain and hold a job. Both social and economic concerns may motivate Congress to take action to improve literacy in American society.
- *New institutional roles:* OTA found that many public educational institutions are under severe strain, to the extent that many question their survival—at least in their current form. Actions directly related to the use of information technology could also have important impacts on these public educational institutions, both by enhancing their productivity and by helping them offer a modern, computer- and communication-based curriculum.

Although the States have primary responsibility for control of the public schools, decisions and policies set at the Federal level have influenced the nature of public education and will continue to do so.

Options for Federal Action

Assuming that Congress decides there is a significant need for Federal action to address these issues, there are a number of possible actions it could take.

- **Direct Intervention**

  Congress could take action to increase and improve the use of information technology in education. Most of the following options would principally affect the schools. A few would have a broader effect on the provision of education and training in other institutions.

  —*Provide tax incentives for donations of computers and other information technology:* H.R. 5573 and S. 2281 are examples of such initiatives. They are intended to accelerate the rate at which schools install computer hardware and to respond to possible inequities in the abilities of school districts to direct funds to equipment acquisition. However, some experts have noted that the personal computer industry is on the verge of moving to a new generation of more powerful machines that may have much greater potential for educational application on a more sophisticated level. Donations of older equipment could freeze the schools into dependency on obsolescent systems. Moreover, such incentives do not address problems such as the need for software, teacher training or institutional barriers to effective use.

  —*Subsidize software development:* OTA found that the most-often cited barrier to current educational use of technology was the lack of adequate educational software. There may be a role for the Government in reducing the risks software producers currently see that inhibit major investment in quality courseware (educational software). Many of the existing successful packages, such as the Sesame Street programs for television and the PLATO computer-aided instruction system, were developed with partial Federal support. On the other hand, good software may be forthcoming if the producers see a sufficient quantity of hardware in the schools to provide them with a viable market.

  —*Directly fund technology acquisition by the schools:* The Federal Government could directly underwrite the acquisition of hardware and software by the schools. Such a program would create a market for educational products that would attract producers, and it would accelerate the introduction of technology into the schools. On the other hand, it may promote premature and unwise purchases of technology by schools that are unprepared to use the technology effectively. Such a program is also counter to some current trends and attitudes in Congress concerning the proper Federal role in education.

  —*Provide support activities:* The Federal Government could assume a leadership role in encouraging the educational system to make more effective use of information technology by funding demonstration projects, teacher-training programs, and the development of institutions for exchanging information about successful implementations. OTA found evidence of a high degree of interest and motivation by both schools and parents that could be more effectively channeled with appropriate Federal leadership. Such a program would not address the financial limitations that currently prevent many institutions from acquiring technology and software.

- **Adapt a General Education Policy**

  Congress is considering various forms of education-related legislation that may affect, and in turn may be affected by, the new informational needs of society. Examples are bills concerning vocational education, veterans' education, education for the handicapped, and foreign language instruction. Such legislation, if drafted with the intent to do so, could encourage the development of more effective and economical technological alternatives to current programs.

- **Support R&D**

  Federal civilian agency support of R&D in educational technology has decreased substantially over the last decade. OTA found that, to make the most effective use of technology, there was a need for R&D in learning strategies and cognitive development, methods for the production of effective and economical curricular software, and the long-term psychological and cognitive impacts of technology-based education. Congress could consider policies to: 1) directly support R&D in these areas, 2) encourage private sector investment from both foundations and industry, or 3) encourage a combination of both by using Federal funding to leverage private investment.

- **Elimination of Unintended Regulatory Barriers**

  Some legislation and regulation not specifically directed at education may create barriers to the effective application of educational technology. Telecommunication regulation, for example, can affect the cost of technology, access to communication channels, and the institutional structure of education providers.

  Moreover, protection of intellectual property, principally copyright law, was identified as a major determinant of the willingness of industry to invest in educational software. The current state of the law was seen by many industry experts as inadequate and, hence, as creating a barrier to the de-

BEST COPY AVAILABLE

velopment of novel and innovative software. However, to the extent that such a barrier does exist, it is not clear whether its removal lies in new legislation or in the gradual development of legal precedent in the courts.

NOTE: Copies of the full report "Informational Technology and Its Impact on American Education," can be purchased for $8.00 from the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402, GPO stock No. 052-003-00888-2.

General Information

Information on the operation of OTA, the nature and status of ongoing assessments, or a list of available publications may be obtained by writing or calling:

Office of Congressional and Public Communications
Office of Technology Assessment
U.S. Congress
Washington, D.C. 20510
(202) 226-2115

Publications Available

OTA Annual Report—Details OTA's activities and summarizes reports published during the preceding year.

List of Publications—Catalogs by subject area all of OTA's published reports with instructions on how to order them.

Press Releases—Announces publication of reports, staff appointments and other newsworthy activities.

OTA Brochure—"What OTA Is, What OTA Does, How OTA Works."

Assessment Activities—Contains brief descriptions of assessments presently under way and recently published reports.

Contacts Within OTA

OTA offices are located at 600 Pennsylvania Avenue, S.E., Washington, D.C.

Office of the Director	224-3695
Office of Congressional and Public Communications	224-9241
Energy, Materials and International Security Division	226-2253
Health and Life Sciences Division	226-2260
Science, Information and Natural Resources Division	226-2253
Administration Office	224-8712
Personnel Office	224-8713
Publications	224-8996

END■

Toward Improving Learning Strategies and Personal Adjustment with Computers

by
Kenneth A. Ryba and James W. Chapman

Introduction

The use of computers for individualizing learning processes has received much attention. Frequently, the advantages of computers are described in programmed instruction terms (i.e. self-pacing, immediate feedback, small steps) with recognition given to the importance of building these principles of learning into educational software. Little, however, is being said about the effects of computers upon the ways students learn and the manner in which they perceive their own abilities. Our objective here is to present some ideas based upon observations and research findings concerning the impact computers may have upon students' personal adjustment and learning styles. The view we advance is that both the cognitive (intellectual, problem-solving) and affective (social, emotional) nature of computer-learner interactions may exert a powerful influence on the type and quality of learning that takes place with any one person.

Recently, interest has been shown in the possibility that computers may change the manner in which children learn to solve problems and process new kinds of information. Enthusiasm has been shown for approaches such as Logo (Papert, 1980), which claim to provide a personalized learning environment in which children are in control instead of being controlled by computers. It is this concept of personal control that keeps appearing in our own work and other research as an important underlying factor. Our own experiences lead us to speculate that feelings of personal control and effectiveness over the computer medium may yield psychological benefits including improvements in self-concept, reduction of emotional dependence and self-management of behaviour and learning. Presented here are some practical suggestions for improving computer learning based upon research findings and educational/psychological theory. Also included is a description of research in progress which we hope will yield more precise information concerning the effects of computer interactions on behaviour and learning.

Computer Games Aid Adaptive Behaviour

The power of highly interactive learning environments is most clearly visible in various computer games and simulations now available. We have watched with interest as mentally retarded students struggle at the outset to understand the "rules" of these games and then advance over a series of trials to master the program and achieve high scores that would in some cases at least make them the envy of any game parlour habitue. Observations of these students' behaviour clearly indicate the development of a range of new skills, including planning, anticipating various events, sequencing left and right, executing combinations of movements (e.g. moving a cursor and "shooting" at targets), shifting attention to different relevant cues, and increased motor speed and coordination. The main point arising from use of these "games" is that much work remains to be done on studying the ways in which people learn and on improving the interactive capabilities of computer assisted instruction in its traditional form.

One important observation we have made is the apparent incongruence between students' performance in the classroom or testing situation versus performance when interacting with computers. Mary Anne, a 25-year-old mentally retarded student, was unresponsive to psychological testing to the extent that she was unable to verbalize or follow instructions as required by the test. Accordingly, she was labelled as severely retarded and unlikely to benefit from formal training. A few weeks later, however, we happened to catch a glimpse of Mary Anne during one of her regular sessions on the computer and noted her obvious on-task behaviour. She was playing a space invaders type game and appeared able to understand the requirements of the task. A closer examination revealed she was able to purposefully interact with the game by moving the cursor left or right to avoid collisions, "firing" at moving objects, and making judgements concerning the order in which operations should be performed.

These wide differences in behaviour underline the shortcomings of intelligence tests which provide only a limited sample of performance. Clearly, Mary Anne's ability level could be severely underestimated on the basis of test results, whereas her actual performance in a real-life situation using the computer indicated a much higher functioning level. The suggestion here is that intelligence may be defined more meaningfully on the basis of adaptive behaviour skills, e.g., purposeful interaction with computers, rather than static descriptions of intelligence, e.g., I.Q. tests. This discrepancy also raises the point that many students could be underfunctioning through lack of opportunity to demonstrate and improve their cognitive abilities with the aid of learning devices such as computers. Such observations add fuel to the case against using standardized paper-and-pencil intellectual assessments.

Goals and challenges imposed by games are ego involving in the sense that a learner is required to evaluate his/her own performance. Success in a computer game appears to bring with it a perception of mastery or ability to control one's environment. Moreoever, this personal control seems to be an internal experience specific to the individual concerned. Most of us are probably all too aware that the motivation to play games is largely an intrinsic process with little or no need for extrinsic rewards such as verbal praise or prizes. Clearly, if it were possible to create other learning environments that contained these elements of intrinsic motivation and personal control, then it might be possible for students to make more rapid progress with their learning.

Developing Internal Control with Computers

The manner in which students perceive themselves and the effects of their own actions upon the external environment is most frequently explained in attribution theory terms (Weiner, 1972). In a basic sense, an attribution is a perception which a person has about the causes of a specific behaviour or outcome. For example, some students may largely explain success on a test in terms of luck or chance happenings (external control), whereas other students account for their achievements on the basis of advance study (internal control). Individuals have been shown to attribute their relative success or failure on a learning task to several factors, including their ability, their effort, the difficulty of a task and/or chance happenings. Ability and effort can be considered as internal since these attributions originate from within the person, whereas task difficulty and luck are attributable to external causes.

In an attempt to explain why CAI has been so successful in the Los Nietos School District, Crandall (1977) advanced the view that computer interactions could be effective for building internal control in those students in whom it is lacking. Studies have shown that low-achieving students and those from minority groups tend to see themselves as dominated by external factors or intervention of people more powerful than themselves. Crandall believed that computers could be used to help these students perceive a causal relationship between their own actions and the feedback that occurs. Specific features of CAI considered to foster this internal control were the lack of external factors to which students might attribute success or failure, one-to-one relationship between the student and computer, and the successful learning experiences resulting from most CAI courseware. Once students become internally controlled, they are more likely to make gains in mathematics and reading skills. Whether or not a student is, in reality, able to exert control over instruction may not be so important as the internal sense or feeling s/he has of being in control. It is this inward state of control which appears to be vital for improving academic achievement.

In our own work with mentally retarded adults, we have observed that many of these people appear to achieve a sense of potency by exercising control over a microcomputer to enhance personal learning. Many of these adults have led sheltered lives and have had few opportunities to act purposefully and take responsibility for their own decisions. For these students, whose lives are controlled most of the time by external agents (workshop supervisors, group home parents, bus timetables, etc.), the computer provided them with a rare opportunity to gain a sense of achievement and self-worth. This was evidenced by students waiting for their supervisors to leave the room in order to proceed independently with their work on the computer without fear of interference from others. Such independent behaviour as this perhaps reflects an attempt by students to exercise control over their learning environment. Paradoxically, while the computer itself generates a large measure of control over the learning situation, individuals tend to perceive themselves as "being in charge" to a greater degree than is customary in a conventional teacher-controlled environment. Furthermore, it would seem that the relatively impersonal nature of a computer can promote high levels of independence or self-management of learning, whereas one-to-one types of interpersonal tuition may foster dependency behaviours between the student and/or student.

"Whether a student is, in reality, able to exert control over instruction may not be so important as the internal sense or feeling s/he has of being in control. It is this inward state of control which appears to be vital for improving academic achievement."

The theory of learned helplessness has significance also for explaining students' achievement in response to CAI. Thomas (1979) offered a definition of learned helplessness as a phenomenon by which an individual learns, over a series of trials, that s/he has no control over the outcome of events. Thus, the helpless individual sees no relation between effort and the attainment of a particular goal. Research by Floor and Rosen (1975) has also demonstrated that rigid, institutionalized lifestyles can contribute to compliance and learned helpless responses. In attribution theory terms it is to be expected that external attributions of control will be associated with learned helplessness. It is interesting then to speculate upon whether the interactive medium of computers can be used to overcome passivity and helplessness by assisting students to see that a causal relationship exists between their own actions and success or failure in learning. No systematic research yet exists to examine this speculation.

The theme of being able to exercise personal control over computers was, however, found in interviews with slow learning high school students. Frequent mention was made of control features available with computer instruction. For example, when asked what they liked best about using the computers, nearly 50 percent of the students mentioned some element of control or decision making, e.g., "I can work on my own"; "I can go at my own speed and level." This suggests that students tend to perceive themselves as being "in charge" of the computer medium and more able to direct their own learning than they would otherwise be able to do in a classroom environment. Perhaps CAI might be used to foster the development of internal attributions such that students monitor their performance and adopt mastery-oriented achievement behaviour. Research is needed, however, to verify this speculation.

Personal Adjustment Training with Computers

Interviews with teachers concerning the usefulness of CAI revealed that the major advantages were seen in social and emotional terms rather than academic performance outcomes (Ryba, 1982). Two case studies in particular serve to illustrate these interpersonal aspects. Martin, a very shy, anxious, nonverbal student found it difficult to communicate with his peers and was subject to ridicule or verbal abuse by other students. Upon being assigned to manage the computer, he appeared to develop more self-confidence as he had an opportunity to demonstrate his technical skills and to assist other students with their work. Martin began to communicate more openly and gained such acceptance by his peers that he was eventually nominated by his class to represent the work experience unit on a school wide committee. Another student who tended to be truant from school much of the time demonstrated a large amount of interest in the computer programs and agreed to contract with the teacher to complete his classwork in exchange for time on the computer. His school attendance improved remarkably when he was given a chance to participate in CAI. Many other examples such as these could be cited to illustrate ways in which computers are used to improve students' personal adjustment. A most salient feature of the computer environment appears to be the opportunity for control or self-management it affords to student who have repeatedly experienced failure with past learning.

Internal attributions and social problem-solving skills may be taught or reinforced through computer assisted learning. In a pilot project at Queen Elizabeth College, Palmerston North, New Zealand, mildly mentally retarded students are receiving a personal adjustment training programme using both group counselling and computer instruction procedures (Young, 1982). In the group situation, students discuss and rehearse problem-solving strategies using real-life events—for example, 1) how to deal with teasing or verbal abuse, 2) appropriate methods of expressing anger, and 3) acceptance of responsibility for one's own behaviour. Following each session, students work in teams analyzing solutions to problems presented by the computer. Some lessons use multiple choice questions to illustrate a cause-effect relationship. For example:

> Q. Bill needed to hurry or he would be late for school. Bill saw Pete on the way to school and stopped to have a smoke with him. Bill got a detention because:
> a. the teacher didn't like him.
> b. stopping for a smoke made him late.
> c. the teacher doesn't like Bill and Pete smoking together.
> d. it is Bill's responsibility to be on time for school.

The computer is used in this way to help foster accurate self-appraisals by requiring students to examine the causes of problems and pose likely solutions or consequences. This approach may in turn assist students to develop internal control and coping strategies.

A research project at Massey University[1] is now being undertaken to determine whether long-term computer instruction can aid in reducing overly passive or learned helpless behaviour. It is hypothesized that highly interactive computer instruction with extensive feedback (using synthetic speech) and learner control will improve self-directed learning and internal control to a greater extent than traditional CAI methods. Participants in the project will be correctional institute and high school students who demonstrate overall high levels of helplessness and/or external control on self-report scales and behaviour checklists. The effects of computer learning upon personal adjustment will be examined by comparing the outcome of highly interactive, user-friendly instruction with conventional CAI methods. This program development is being implemented on a New Zealand microcomputer system called POLY 1 which makes use of multiple terminals, a network controller and an 8" disk drive.

---

[1] Results from this study will be available later this year. Contact Ken Ryba for details.

Computer Training of Cognitive Processes

Discussion up to this point has focused on affective characteristics concerning the extent to which students modify their behaviour through interaction with computers. Now we will briefly consider some cognitive-developmental changes that might also be occurring with computer learning.

Increasing attention has recently been given to using computers to aid the development of learning styles and cognitive strategies. In describing Logo, for example, Papert (1980) points out that by controlling the computer, children embark on an exploration of their own thinking processes. He suggests that through experimentation and self discovery, learners can become active builders of their own intellectual structures. Whether or not these speculations are shown to be true, it seems likely that approaches which emphasize active learner participation will be most effective.

In our own research we have observed high levels of active responding and "on task" behaviour. Systematic observations of computer use by slow-learning high school students revealed that, on the average, students spent 91% of their total time actively engaged in attending and problem solving. Related to this, it was noted that students frequently made self commentaries concerning their interactions with the computer (14% of total time). It is known from previous research (Brown, 1975) that overt verbalizations of this type can significantly enhance the performance of slow learners. This finding suggests that the interactive nature of computer learning might be helpful for mediating problem-solving processes, especially with students who are unresponsive to other more passive instructional media. As such, well-structured computer learning activities could help organize input materials for rapid processing by the student. In effect, when a teacher tells children to be quiet, s/he may really be saying to them, "Stop thinking."

Also interested in studying conditions for learning, Jewson and Pea (1982) observed that primary school children seem to collaborate and teach each other more when they are working with microcomputers than they do in other classroom work. The results of one study indicate that children talk more to each other about problems they are doing when using a computer as opposed to other conventional class assignments. Jewson and Pea make the point that the quality and occurrence of such collaborative work are important aspects of the computer learning context.

Such findings as these highlight the importance of providing flexible and personal learning systems rather than highly structured CAI systems. For example, Logo, instead of CAI, could be used to teach young children such basic operations as number concepts, word recognition, directional concepts and logic. This would enable teachers to use their preferred teaching styles while engaging children in personal exploratory learning.

A theory of attention by Zeaman and House (1963) proposes that many types of learning difficulties are characterized by problem with information processing. Specifically, it has been shown that mentally retarded students experience difficulty sorting out relevant from irrelevant cues. However, once these individuals learn to attend to relevant aspects of a task, their learning proceeds in a manner which is similar to that of a "normal" learner. Assuming these findings can be upheld, we are left wondering how often educational software is made ineffective through the introduction and reinforcement of irrelevant bits of material. Unfortunately, there is so much fascination with special effects, e.g., sound, high resolution graphics, that the educational benefits (if any) are often destroyed through inappropriate stimulation. Courseware developers who do not have a background in educational theory should especially take note of this point.

Another study we are undertaking at the present time concerns computer aided training in problem-solving skills, planning ability and motor speed/coordination. Participants in this project are mentally retarded adults attending a vocational training centre. A group of tests was administered individually to each person in order to assess such cognitive abilities as recall of words and digits, gross and fine motor coordination, copying figures and identifying missing parts of puzzles. Participants were randomly assigned to experimental and control groups with 30 people in each group. Computer training for the experimental group was a game procedure (*Galaxy Invaders*) which is thought to be a rich environment in which to learn relevant cues and cognitive strategies. Control group members were also given a game program (*X's and O's*), though this procedure required relatively fewer strategies than the experimental program. Following completion of twenty training sessions, each person will be retested. Additionally, transfer tests will be used (a computer game, assembling a transformer, threading a loom) to examine whether or not learning which takes place in the computer situation can be generalized to other more crucial real-life events. We hope that this type of applied research will provide a basis for more specific studies concerning the use of computers for training cognitive processes.

Finally, one is perhaps left wondering how it will be possible to translate all this theory into practice. A great deal of helpful information unfortunately never finds its way from research journals into classrooms. While there are no easy answers to this, it is reasonable that as new knowledge becomes available, it should be captured in software design.

Guidelines for Translating Theory into Software Design

How can we use theoretical results now available to make more effective learning programs for students? We think the characteristics that have been discussed here can easily be built into much software. Following are some of the main points arising from this review of theory along with examples of ways in which each recommendation can be put into practice. All of these guidelines apply equally well to noncomputer teaching situations.

GUIDELINES	EXAMPLE
1. **Maximize student control.** Ensure that students can direct the program at several points with decisions such as to advance, quit, repeat, review, etc.	"Did you understand this module?" "Would you like to review again?"
2. **Personalize Internal Feedback.** Use messages that reinforce the student's personal effectiveness.	"You learned this work really well." "Congratulations on achieving a higher score than last time. This was a difficult assignment and you did a fine job."
3. **Self-Evaluation.** Provide opportunities for the student to rate his/her own progress.	"Did you have any difficulty with this?" "What should you do now? 1. Study some more? 2. Try harder? 3. Go to the next module?"
4. **Goal Setting.** Provide opportunities for the student to set personal goals.	"Last time you had 5 correct answers. How many will you get this time?"
5. **Evaluation of Goals.** Ensure that the student assesses his/her performance on the basis of previously set goals.	"Did you reach your goal? Was your goal: 1. Too high? 2. Just right? 3. Too low?"
6. **Personally Meaningful Rewards.** Provide an opportunity for the student to contract for type of reward and conditions for achieving reward. Note that this behaviour contract could be negotiated directly between student and teacher as well.	"Set your contract here: . . ." "How many points earn reward (minimum is 10)?" Select type of reward: 1. Game 2. Free period 3. Extra computer time."
7. **Stress Relevant Concepts.** Ensure that the program clearly displays essential information needed to complete a task. Use prompts to focus students' attention on important elements.	Use colour coding to highlight grammatical terms. Avoid excessive use of graphics unless they are relevant for learning a specific operation.
8. **Encourage Self Commentaries.** Research has shown that overt verbalization can help mediate many kinds of problem solving. Collaborative learning could be encouraged by assigning teams of children to each computer.	"Now you say this number." "Say what colour this is." "Show another student how you solved this problem."

END■

REFERENCES

I. Vocational and social training. In K. (Ed.) *Orientations in special education.* rk: John Wiley & Sons, 1975, pp. 159-198.

N.D. *Computer assisted instruction: How children's achievement scores.* Internal Los Nietos School District, California.

: Rosen, M. Investigating the phenome- elplessness in mentally retarded adults. *n Journal of Mental Deficiency,* 1975, *79,* 572.

, and Pea, R.D. Logo research at Bank ollege, *BYTE,* 1982, 7(8), pp. 332-333.

ymour. *Mindstorms: children, computers, erful ideas.* Brighton: Harvester Press,

a. Evaluation of computer assisted in- n with work experience classes. Unpub- esearch report, Massey University, Pal- North, New Zealand, 1982.

Thomas, A. Learned helplessness and expectancy factors: implications for research in learning disabilities. *Review of Educational Research,* 1979, 49, pp. 208-221.

Weiner, B. Attribution theory, achievement motivation and the educational process. *Review of Educational Research,* 1972.

Young, V. Computer aided personal adjustment training. Unpublished research report, Queen Elizabeth College, Palmerston North, New Zealand, 1982.

Zeaman, D., and House, B. The role of attention in retardate discrimination learning. In N.R. Ellis (Ed.) *Handbook of mental deficiency.* New York: McGraw-Hill, 1963, pp. 159-229.

Kenneth A. Ryba and James W. Chapman, Department of Education, Massey University, Palmerston North, New Zealand.

Toward a Computer Literate Society— An Elementary School Responsibility

by
Kay Walla
Assistant Chair and Associate Professor
Department of Teacher Education
Director: Reading/Learning Center
William Holmes McGuffey Laboratory School
Miami University
Oxford, OH 45056

Virginia Brubaker
Teacher—Gifted
Indian Hill School District
Cincinnati, OH

"Computers directly affect the life of every person living in this country, therefore a literate person today and for the future needs to know how the computer works and the pervasiveness of its influence." (MECC, 1980) It is pedagogically sound to make the computer a part of the learner's academic environment so that it becomes a recognized and accepted part of the total environment and loses its image as a strange, unknown or perhaps bewildering machine.

Teachers in the William Holmes McGuffey Laboratory School, hearing the message and sensing the need in the Spring of 1981, developed a prototype curriculum for enhancing computer literacy. The subjects for the implementation of the computer literacy study were 30 nine-year-old (fourth grade) students in the William Holmes McGuffey Laboratory School (Miami University, Oxford, Ohio). The objectives of the six week unit included:

1. Recognizing and using computer vocabulary.
2. Learning about functions, devices, capabilities and limitations of computers
3. Being aware of the speed and accuracy of computers
4. Understanding the widespread use of computers in today's society and the impact of computers on our daily lives
5. Learning that people control computers
6. Projecting future computer applications and the potential impact computers may have on the students' lives
7. Exploring the historical development of computers
8. Designing simple programs

The Design

Six microcomputers, 4 Apples with single disk drives and color monitors, and 2 TRS-80 Level II Model I with tape recorders were housed in the fourth grade classroom for six weeks. The students were engaged in computer literacy activities for at least three forty-five minute blocks per week. Two or three students per monitor kept half of the class occupied at a microcomputer station with one teacher assisting, while a second teacher worked with the remainder of the group in pre or post computer activities. Many students also interacted with the microcomputers in their spare time both during and after school. A schedule on the teacher's door indicated times when teacher supervision was available and students could schedule 1/2 hour blocks at their leisure. The teachers encouraged follow-up activities and applications in other curricular content areas. Each student completed a learning packet on computer literacy designed by teachers to include vocabulary activities, computer history and worksheets which corresponded with programs they ran. Students selected programs and related print packets in at least one curricular area. Their choices included programs in health, spelling, math, library skills, social studies and English. These programs were selected from the Minnesota Educational Computing Consortium Series, Micro Power and Light, Milliken, Hartley Software, Edusoft and Ideatech. (The students' favorite programs were science and social studies simulations from MECC.)

Students assisted in the management of the curriculum by keeping their own files and recording in their packets the names of programs they completed. Teachers played the role of facilitator, supervising students on microcomputers as well as those who read books,

BEST COPY AVAILABLE

played computer games, developed scrapbooks and practiced the keyboard on two manual typewriters while anxiously awaiting their turn at the computer. Teachers also designed and administered pre and post assessment in an effort to gain some sense of growth of the students' computer awareness and to assess the value of the unit. Books, films and magazine articles were ready references in the classroom for both the students and their teachers (see References). Additionally, interested adults from the business world, including a few parents, stopped in to see the youngsters at work and to talk with them about computer applications in their jobs. All adults were impressed by the children's interest in and ability to use the computers.

> ***"While a concept to be presented may be that computers are used to speed up or simplify work, students need to understand that programming a computer takes a lot of time. It could be perceived as an inconsistency that computers work quickly while the student is actually experiencing a slow and tedious process."***

Two field trips were taken to Cincinnati-area industries. Procter and Gamble and Cincinnati Milacron employees demonstrated computer applications, including the use of graphics in product design and the use of an industrial robot.

Reactions and Implications

An investigation of pre and post test scores indicated a great deal of knowledge gained in areas such as what a computer looks like, what it does and does not do, and ways it can be used. A computer vocabulary was gained not only through the use of programs and written activities, but through use by kids who quipped such things as, "I don't compute!" Scrapbooks were minimally started in this project, but early entries indicated ideas concerning the widespread use of computers in our daily lives. Perhaps the greatest evidence of learning that people control computers was the frustration that occurred when students did not follow directions or when simple programming began and students realized the explicitness required for the computer to perform the smallest function.

Some things were learned in this unit which would be helpful to other teachers who plan a similar computer literacy unit:

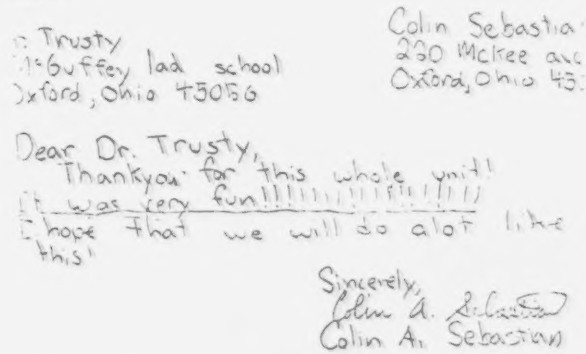
1. Teachers need to be specific in their emphasis on the speed of computers. While a concept to be presented may be that computers are used to speed up or simplify work, students need to understand that programming a computer takes a lot of time. Children's lack of skill with a keyboard makes inputting data an even slower process. Use of commercially published programs can also be slow, depending upon the reading skill of the user. It could be perceived as an inconsistency that computers work quickly while the student is actually experiencing a slow and tedious process. In addition, students need to realize that the accuracy of the computer is dependent upon the accuracy of the programmer.
2. Teachers should provide adequate clock time to prevent frustration among students who feel they never get enough computer time. Our schedule of three days a week for forty-five minutes was not enough. Limited by six computers for twenty-eight children, and plagued with occasional technical difficulties, children were frustrated by delays and turn-taking. Three days a week for an hour each would be more realistic for middle grade students. Pairing two students is also advantageous, both for time efficiency and for peer assistance for slower readers or thinkers.

> ***"Teachers should be willing and alert to revising objectives and activities as students' reactions suggest. The computer graphic demonstration at Procter and Gamble was seen by the youngsters as a mere sideline because they were so excited about the electronic mail cart which passed them in the hall!"***

3. Teachers should use the microcomputers for ongoing activities throughout the year. However, if they are not available full time, a nine week unit would be more appropriate than six weeks. Children need time to "figure out" what's happening. At this age some aspects of computer use may not be physically or conceptually easy. The six week unit barely whetted the appetites of the youngsters and hardly allowed a start for activities such as scrapbook collections of computer articles, experimenting with new applications and designing programs.

4. Teachers should be willing and alert to revising objectives and activities as students' reactions suggest. For example, a collage of computer applications was suggested as an alternative to the scrapbook idea. Or, for another example, the computer graphic demonstration at Procter and Gamble was seen by the youngsters as a mere sideline because they were so excited about the electronic mail cart which passed them in the hall! Neither their tour guides nor their teachers were prepared to switch gears and capitalize on the keen interest they showed in an obvious application of computer usage.

The positive aspects of this unit cannot be emphasized enough. Not only were children excited throughout the unit, but for the weeks and months that followed they competed for computer sign-up time before and after school as well as during the summer months. These are nine year olds turned on to a reality of the present and a necessity of the future. Faculty and parents, not to be outdone by the students, got into the act—some as users, some as programmers and some as purchasers. Enthusiasm ran rampant throughout the unit and children in other classes placed pressure on their teachers to provide *them* with computers in *their* classrooms. One student very eloquently represented the attitude of his classmates in the following thank you note—



Colin Sebastia
230 Mckee avc
Oxford, Ohio 45

r. Trusty
1cGuffey lad school
xford, Ohio 45056

Dear Dr. Trusty,
Thankyou for this whole unit!
It was very fun!!!!!!!!!!!!!!!!!!!
I hope that we will do alot like this!

Sincerely,
Colin A. Sebastian
Colin A. Sebastian

This letter illustrates the enthusiasm of an actively motivated young student, and all of the faculty involved support Colin's hope that "we will do a lot like this!" It is incredibly shortsighted to exclude computer literacy from the elementary school curriculum. Units are already incorporated in many schools. Efficient models are available, and software is now forthcoming from publishers who realize the demand and teachers who realize the capabilities of this instructional tool. The interest is spontaneous with young people who are fascinated with the sophistication of the computer and unafraid of its power. The content is relevant and motivating. The microcomputer makes computer literacy both financially possible and educationally convenient to implement. The computer is **NOW** as well as in the future. Let us not take years to recognize its pervasiveness and to incorporate it into the curriculum. "The illiterates of the next generation will not be those who cannot read—but those who cannot program computers and use their capabilities." (Gambrell and Sandfield, 1979)

[Ed. note: For further information about this project, contact Dr. Walla at the above address.]

STUDENT RESOURCES:

Broeder, Michael, *Larry Learns About Computers*, T.D. Denison and Company, Grades 1-3.

Burger, *Computers (Science is What and Why)*, E. M. Hale and Company, Grades 1-3.

Clark, John E. *Computers at Work*, Grosset and Dunlap, Grades 4-6.

Kenym, Raymond, *I Can Learn About Computers and Calculators*, Harper and Row, Grades 4-6.

Rae, Joanne, *Careers in Computing*, E. M. Hale and Company, Grades 1-3.

Rice, Jean, *My Friend the Computer*, T. S. Denison and Company, Grades 4-6.

TEACHER RESOURCES:

Braun, Ludwig, "Computers in Education," *Apple*, Vol. 1, No. 1, Apple Computer, Inc., Cupertino, California.

______________, *Creative Computing Journals*, Morristown, New Jersey, monthly publication.

Gambrell, James and Sandfield, Robert, "Computers in the School: Too Much Too Soon?" *High School Journal*, 62:327-31, May 1979.

Jornir, Lu Marvin, Miller, Sidney R., and Silverstein, Burton J., "Potential and Limits of Computers in Schools," *Educational Leadership*, ASCD, March 1980, p. 498-501.

Mason, George and Blanchard, Joy, *Computer Applications in Reading*, International Reading Assoc., 1979.

Milner, Stuart D., "Teaching Teachers About Computers: A Necessity for Education," *Phi Delta Kappan*.

Rice, Jean, *My Friend the Computer*, T. S. Denison and Company, Minneapolis, Minn., 1976.

REFERENCES

D'Ignazio, Fred and Gilliam, Stan, *Katie and the Computer*, *Creative Computing*, Morris Plains, NJ, 1980.

Educational Technology, October, 1979 issue, Microcomputers in Education and October, 1981, Computers in Schools, Englewood Cliffs, NJ.

Gambrell, James and Robert Sandfield, "Computers in the School: Too Much Too Soon?" *High School Journal*, 62:327-31, May 1979.

Jornir, Lu Marvin, Miller, Sidney R., and Silverstein, Burton J., "Potential and Limits of Computers in Schools," *Educational Leadership*, ASCD, March 1980, p. 498-501.

Kasel, Marge and Carleston, Geraldine, *It's Elementary, My Dear Computer*, Minnesota Educational Computer Consortium 25, St. Paul, MN, 1980.

Larsen, Sally Greenwood, *Computers for Kids*, A Basic Programming manual written just for kids, Apple or TRS-80, *Creative Computing Press*, P.O. Box 789-M, Morristown, NJ 07960.

Matthews, Walter, "Computers in My Classroom?" *Apple*, Vol. 1, No. 1, Palo Alto, CA, 1979.

Moursund, David, "Introduction to Computers in Education for Elementary and Middle School Teachers, Chapter 8," International Council for Computers in Education, University of Oregon, Eugene, OR 97403, 1981.

Murphy, Edward J., "Developing Computer Literacy in K-12 Education," *The Computing Teacher*, November 1981, pp. 43-48.

Tallant, Dale M., "A Computer in the Classroom!" *The Computing Teacher*, December 1981, pp. 11-13.

END■

BEST COPY AVAILABLE

The Computing Teacher September 1982

The Intelligence Amplifier: An Essay on the Design of Computer-Based Curricula

by
Michael Moshell
Associate Professor of Computer Science
University of Tennessee
Knoxville, TN 37916

Editor's Note: This is Part III in the series which began in Vol. 9 #8 (April 1982) on Computer Power, *an introductory programming curriculum for junior high, high school and junior college students.* Computer Power *was developed in 1979 at the University of Tennessee under a grant from the National Science Foundation. Now in the Evaluation Phase of the project, student performance is being formally evaluated in ten schools.*

Computer Power is a one-semester course, and is now for sale from Gregg/McGraw-Hill. A second-semester course is in preparation.

This paper describes insights and attitudes toward computer-based learning which have developed during the design of the *Computer Power* programming curriculum. Two previous papers in this series have described the curriculum and the underlying software. We begin with a section discussing some underlying concepts and terminology about how learning works.

THOUGHTS ABOUT LEARNING AND THINKING

Syntax and Semantics

A language is a set of symbols, together with two interrelated sets of rules: the *syntactic* rules, which describe the legitimate sentences of the language, and the *semantic* rules, which describe the way meaning is conveyed by sentences. Syntactic rules are necessary to distinguish sentences from the rest of the universe; thus, part of syntax is a description of the alphabet used. Syntax also protects the language against injection of random bits of the universe into sentences, by allowing the listener/reader to detect errors.

To make explicit a language (but *not* merely use it), one must explain the syntax, at least, in another language. This language of explanation is called *meta* to the (object) language under discussion.

When we teach computer programming languages we almost invariably use English as the metalanguage to explain, for instance, Pascal's syntax. We find it harder, however, to use our metalanguage to explain the *semantics* of the language. This is largely because "what a program means" is a series of *actions*—the kind of actions expressed by a flowchart, a movie or a ballet. English is quite poorly suited for representing these phenomena. We have learned that we can't just teach students about flowcharts and expect they will then understand a program's semantics, because this just gives them two new languages instead of one to learn. (It does help some people, however; those with highly visual approaches to reality find that flowcharts are much easier to grasp than programs.) But the reality is that we must teach programming languages much like we teach children natural languages; that is, we must teach the "semantic domain," the set of things the sentences may refer to, simultaneously with the structure of the sentences. (A graphics-oriented computer with immediate mode command interpretation is a good example.) Metalanguage, a language for ex*plaining* what is happening, will play at most a minor role compared to direct experience in the teaching of children.

Despite these caveats against too much faith in the English metalanguage for getting information to students, we shall see that the idea of a metalanguage has important applications when we discuss the emotional environment of the student.

Learning and Meta-Learning

The entire organism of a child is designed for learning. The manifestation of that predeliction is called "play." A school's requirement that children play only rarely and in appropriate contexts is a crucial piece of meta-learning (learning about learning) for the kids. They learn that society cannot afford for them to spend all their time learning, according to their own agendas of interest. Most of their adult lives will be spent delivering goods and services within a social framework that cannot tolerate too much innovation. We are obliged to teach the following rule about learning: you will not always be permitted to learn or to create. It has been said that the most important thing kids are taught is how to sit in a desk and be quiet and look busy, for that's what they will probably have to do when they grow up.

Meta-learning as a concept encompasses much more than "discipline." We must understand that socialization, play, learning styles and creativity are all part of an omnipresent, implicit ethos. There is a meta-curriculum, a sea of feeling, style and motivation (which we teach) whose currents often run strongly in opposition to the supposed goals of the subject curriculum.

The nature of this meta-curriculum is deeply tangled with the nature of implicit and explicit knowledge.

Let us explore this distinction before pursuing the meta-curriculum further.

Knowledge as Construct: Implicit and Explicit Knowledge

Central to the Piagetian insight into learning is the idea that one cannot be given an understanding; no matter how good the teacher, an understanding must be constructed "on the site," largely of local materials by the natives. I have an image of an optimistic teacher delivering a factory-built prepackaged treehouse kit to a kid, who looks it over and then selects the interesting boards and nails to integrate into her own pile of treasures in the back yard.

Seymour Papert's excellent book *Mindstorms* (1980) describes his childhood experience with mechanical gears, and how their functioning served as a metaphor whereby he understood ratios, linearity and many other mathematical abstractions. He calls this nonverbal, experience-based understanding "affective learning."

> ***"I have an image of an optimistic teacher delivering a factory-built prepackaged tree house kit to a kid, who looks it over and then selects the interesting boards and nails to integrate into her own pile of treasures in the back yard. . . . Play is the behavior of children building their own semantic nets, trying them out, adjusting roles and dynamics."***

This "semantic network" of understanding about treehouse building, gears, programming or whatever, is undeliverable via English, or indeed any language where the kid must sit still. Play is the behavior of children building their own semantic nets, trying them out, adjusting roles and dynamics. And standing squarely across the path of the child, carefully rationing out play-time like fish to porpoises being trained, is the teacher.

Only rarely do we teachers have the luxury of contemplating our complex role in the life of a student. We are conducting, of course, two curricula, two courses of instruction at once. On one side of the report card is reported the "lesson" results: a "C" in geography. On the other side, a note that Maryann "needs to spend her time more wisely," which may mean that Maryann was passionately devouring a book about flowers when it was time for world geography.

> ***"It has been quite inconvienient to bring much of the world into our schools. Many schools teach years of science without a single experiment. Almost no schools use 'real' experiments in which the result is not foreordained and 'rigged.' The central promise that computers hold for education, all of education, is the provision of a domain in which kids don't have to take someone's word as to what happens."***

We take it for granted that most children construct working, effective, comprehensive strategies for dealing with the social pressures of being human. Almost everyone is willing to acknowledge by now that this socialization is taught mostly by nonverbal, subconscious means. No one pretends that Civics class had much to do with structuring the kids' interactions. Yet we persist in believing that the "lesson" curriculum, the one we spend 99% of our conscious energies on, is being transmitted via language from mouths and printing presses into the eyes and ears of sitting children.

We need to understand the relationship of verbal knowledge ("Maryann, what is the capital of Egypt?") to an "affective," creative, dynamic understanding of the world. It is exactly that relationship between Civics and the "behavior curriculum": the verbal component is mostly superficial, taxonomic, and forgettable unless it denotes phenomena within kids' experience.

It has been quite inconvenient to bring much of the world into our schools. Many schools teach years of science without a single experiment. Almost no schools use "real" experiments in which the result is not foreordained and "rigged."

The central promise that computers hold for education, *all* of education, is the provision of a domain in which kids *don't have to take someone's word* as to what happens. "Take someone's word": accept verbal input as the primary source of knowledge. The computer exhibits complex, reproducible behavior cheaply, controllably, and in ways that are very conducive to play. With this tool it is now possible to structure a child's understanding of a subject domain in the proper cognitive order: implicit knowledge ("how-to") *first;* descriptive/verbal/explicit knowledge *after.* The semantics of their growing language (of whatever subject) is not left dangling like the roots of plants in a hydroponic tank, all (temporarily) green of verbiage but devoid of creative intuition.

The well-known *Hammurabi* simulation game teaches children the dynamics of population and food in a play-like setting. As the king, the student must decide how much land to plant with what crops. People starve if wrong decisions are made. After the experience, after the building of a mental version of Babylon, students have something upon which to hang terms such as "investment."

In the *Computer Power* curriculum, little emphasis is given to knowing the names of things. Maximum emphasis is given to showing students how to evoke interesting behaviors of the computer, and to exploring those behaviors for themselves. Making the knowledge explicit is a later goal.

Now we will use the terminology we have developed. We will explore the linguistics and metalinguistics, the curriculum and meta-curriculum, the implicit and explicit learning we have tried to build into *Computer Power.*

COMPUTER POWER META-GOALS

Anxiety and Attitude

Imagine a telegraph set connecting New York and Chicago with this novel organization: there exist two parallel systems, one labelled "statements" and the other "how to take this statement." On the "statement" wire one might transmit, "Who was that lady I saw you with last night?" On the "how-to-take-it" wire we send the message, "Joke is coming"; or perhaps the message "serious question."

> ***"The meta-messages are many and crucial, and the first is: you can do this stuff as well and beautifully as everyone else. We try to continually provide students with visually striking results, to reinforce this sense of well-being and control as the curriculum progresses."***

The how-to channel contains messages *meta* to the statement channel, because the how-to messages contain guidelines for the interpretation of the statements.

Transactions between humans are precisely like this: the non-verbal, postural, intonational channel is constantly telling the reader how to interpret the string of words. A teacher can (if s/he knows how) transmit strong soothing meta-messages to help a student deal with the anxieties of new knowledge. The meta-message is part of a dialogue, with the student's non-verbal (sometimes verbal) message often saying, "I am stupid and unworthy because I don't understand this."

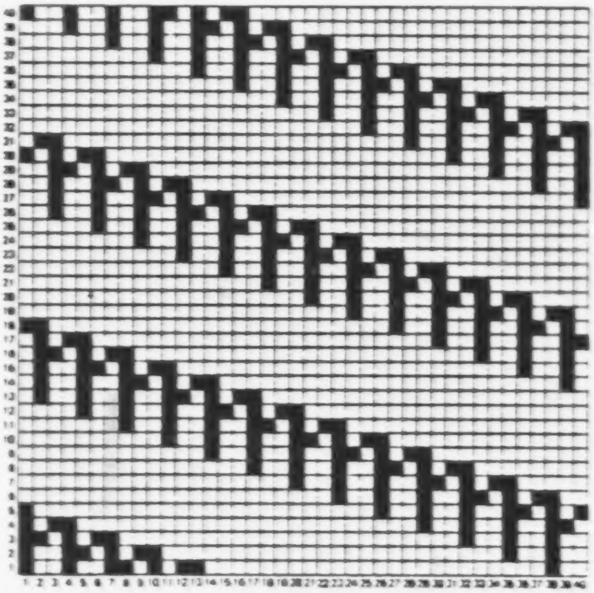
The design of *Computer Power* draws on our experience of teachers; we have tried to provide guidance to future teachers who will be anxious themselves the first time they teach this material. We specifically put the kids in a strongly play-oriented environment at first, with very simple rules. "Quilting" (Lesson 1) is comprehensible to first-graders, yet enjoyable for adults. There is practically no way to fail in producing striking, interesting patterns with the Quilt program. The "message" in Lesson 1 is almost trivial:

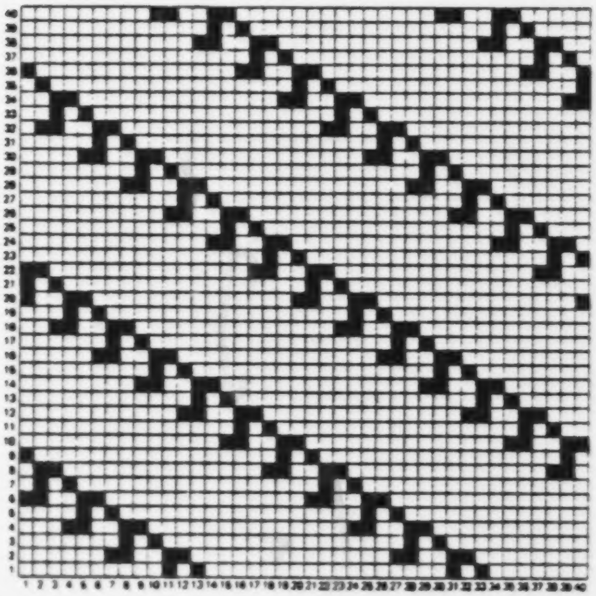
- the eight keys around 'K' move the dot;
- the knob marked '0' changes the color of the dot;
- a certain key causes the pattern to be repeated.

The meta-messages are many and crucial, and the *first* is:

- You can do this stuff as well and beautifully as everyone else.

It's *almost* a con-game, because the beauty proceeds from symmetry and repetition rather than a student's creativity per se; but the observed effect on students is so compelling that we are quite confident that this first meta-message is clearly received. We try to continually provide students with visually striking results, to reinforce this sense of well-being and control as the curriculum progresses.





Discovery Skills

A second component of the meta-curriculum is to foster discovery skills. This also begins with Lesson 1. Students are asked to discover how many squares there are on the screen; how many colors there are; how fast the program draws squares, etc. Answers are available but the teacher is instructed to ask leading questions in response to questions, rather than supplying the answers. Quilting (and successive activities) are so simple at the *subject* level that many of the discoveries take literally just seconds to make. We continually ask the students to discover the rules and parameters of the system, giving as little verbal, *a priori* information as possible. We hope that evaluation will show that students are capable of transferring this ability and willingness to explore to other subjects.

Learning by Experience

A paraphrase of this goal might be "willingness to learn from metaphor." Our Pascal interpreter, INTERPAS, "scrolls" the text of the program in a screen window while executing the corresponding graphics animation in another screen window. Each medium "explains" the other; the programming language statements each correspond with a change in the picture.

The operation of a program is much like Papert's gears; it is a powerful metaphor by which many casual sequences in the world may be illuminated. In our "Blue-Sky" activities involving motion (the Painter/Lunar Lander/Space Ship series), a structured walk-through leads the student to understand velocity and acceleration as mentally visible, moving phenomena. The words which go with the lessons may have some utility *after* the student has realized that (in Painter, for instance), a figure in motion keeps on moving unless disturbed—just as the computer keeps on going around a loop unless something happens to end the loop.

Perhaps what we mean by "experience" here is in fact a rather political goal: we intend to create a need or desire on the part of students for "hands-on" knowledge. Having taken our course, a student will perhaps be less likely to regard the lab component of a chemistry course for instance, as merely smelly "cookbook" production of desired results. It may also increase the dissatisfaction of students with their traditional curricula.

Attention Focus

"Hand simulation" of any formal activity, including long division, requires a quality of attention demanded almost nowhere else except perhaps in chess. From the first lesson we require students to "pretend they're the computer" and to produce graph-paper simulations of expected patterns. Any increment in this ability will serve the student well in future mathematics courses, and perhaps in many other highly structured activities such as writing.

Expressive Skills

Almost at the last of each lesson, we begin to ask students to create patterns or cartoons using what they have learned. They have been given tools which are quite biased toward nonverbal means of expression: music and colored pictures. Some text output is possible, but is rarely central to the lesson.

The experience of creating a cartoon is, nevertheless, comparable to writing an essay (perhaps in a foreign language). One must have a master plan, a command of the syntax and semantics of the language, and an iterative (try and retry) development activity.

While we have no intention of turning *Computer Power* into a composition course, it is also possible for teachers to elicit written answers as part of each weekly quiz. Students then attempt to verbalize what they have learned. We regard the supplying of vocabulary items to the students as primarily serving this goal—to give them something with which to describe their knowledge *after* they are firmly in command of the computer.

COMPUTER POWER SUBJECT MATTER GOALS

Finally, after all this discussion of how we want them to *learn how to learn*, what do we actually want them to learn?

The subject matter curriculum is not, in fact, just a convenient "carrier" upon which the meta-curriculum rides. The things we teach overtly are the core of knowledge that any proficient computer user must know, whether s/he intends to actually write programs or not.

Formal Processes, Programs

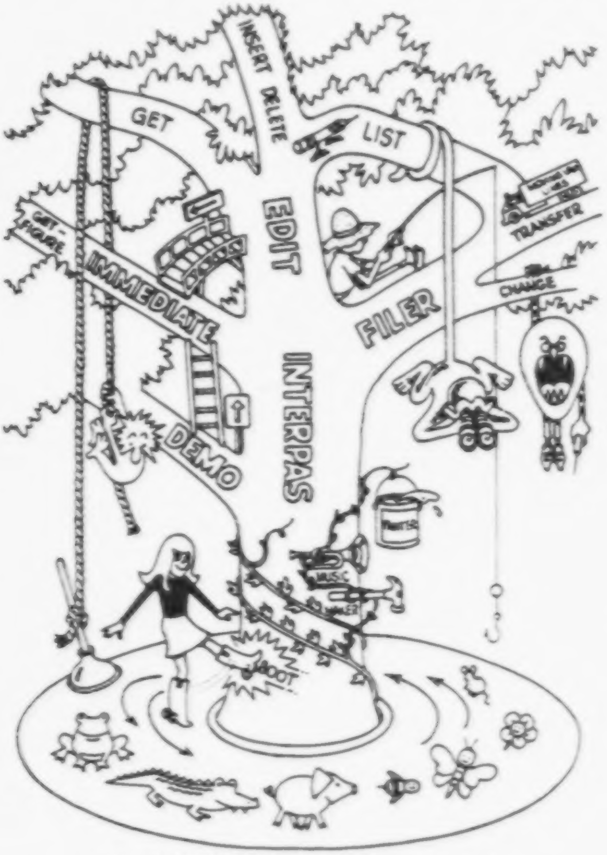
The concept of a rigidly spelled-out, "automatic" sequence of events is perhaps only encountered by precomputer kids in two contexts: home economics and auto repair. In home ec, students are taught to follow a recipe; in auto repair, to follow a repair manual. However, since almost all the words in those two "formal processes" descriptions refer to visible items or actions, they do not adequately train students to deal with invisible abstract processes.

Nevertheless, many such processes exist. In fact, the entire mechanistic notion of nature is hinged on the idea of such processes. One could argue that when "programs" are understood and form the foundation for a child's epistemology, a fundamentally different civilization will have begun (perhaps not very pleasantly, either, if an authoritarian political system were to exploit the metaphor).

Clearly the computer forces its user to understand the nature of explicit sequences of instructions, with no intervening goal-oriented (human)mediator to understand what you *thought* you said. Knowing what a program *is* must be a fundamental piece of understanding for any computer user.

Procedures; Hierarchies of Subtasks

Knowing about programs does not imply, however, understanding procedures. The essence of that assertion constitutes the "structured programming" revolution in computer science. In *Computer Power*, we teach our kids about their first control structure immediately after straight-line code. We teach them to design programs by "making up verbs" that do what they want; then considering the sub-tasks of each verb, down to the primitives of the language. By approaching the teaching of programming in this way, we intend for our students to imagine programs as basically tree-structured from the outset. We teach the control hierarchy of INTERPAS with a rather fanciful tree diagram, showing that it also consists of various procedures, managing sub-tasks.



The INTERPAS informal tree diagram.

> ***"Clearly the computer forces its user to understand the nature of explicit sequences of instructions, with no intervening goal-oriented (human) mediator to understand what you thought you said."***

KNOWLEDGE OF A REAL COMPUTER LANGUAGE

Students in *Computer Power* learn to program in a dialect of Pascal. The only variations from the originally defined language are:

- one statement per line; omission of semicolons;
- addition of numerous primitives ("verbs") to mediate graphics and input-output.

Students are required to declare all variables; to use control structures such as procedures, REPEAT-UNTIL, FOR-DO and BEGIN-END. They learn to declare arrays and to use data types INTEGER, BOOLEAN and a graphics type COLOR.

SUMMARY

We hope that this paper's presentation of concepts, in pairs like Noah's animals, will spark thought and discussion among computer-curriculum developers, particularly those of us who intend to teach programming. For a final encore, here they are once more, albeit slightly reorganized:

- We must think carefully about the relationship of words and action in teaching. The computer provides opportunity to redress an ancient imbalance that essentially denies *fun* its proper place in learning by sanctifying verbal behavior at the expense of nonverbal behavior.
- We must constantly teach at two levels: mastering subject matter, and improving learning skill. The latter is a predominantly emotional domain because it involves taking risks.
- The computer is a unique tool for creative curriculum design because it can allow the child to chart her/his own course between the verbal and nonverbal worlds; between subject matter and meta-learning . . . but *only* if the student is in control, and "play" is legitimate. END■

BEST COPY AVAILABLE

Should Young Children Work with Microcomputers—Logo Before Lego™?

by
B.J. Barnes and Shirley Hill
Division of Teacher Education
California State University, Fullerton
Fullerton, CA 92634

Editor's Note: B.J. Barnes and Shirley Hill are both Professors of Education at California State University, Fullerton. In addition, Shirley Hill is Coordinator for the Microcomputer Resource Center for the School of Human Development and Community Service at the university.

After reading this article we are convinced that children in the primary grades can enjoy and learn from "preschool" software. We also expect that this article will provoke healthy discussion and research activity.

These years of transition from the industrial to the information age are exciting times and we are glad to be living in them: Certainly children must become comfortable with the tools and processes that impact their lives. Their comfort depends upon their developing the understanding, the knowledge and the skills fundamental to the world's increasing complexity and the interdependency of its peoples. In this age of instant communication, but limited understanding, we have access to a marvelous new tool—the microcomputer.

> ***"... a preoperational child would not be harmed by contact with a computer. However, experiences with a computer should never replace their experiences with real events and objects."***

Public schools have always had as their mission the promotion of communication through the development of literacy. Today, the definition of literacy must be expanded to include "computer literacy." Just as we want all children to be able to read books and newspapers critically, to write coherently and thoughtfully, and to calculate accurately, we also want all children to use the computer effectively.

Although we are focusing here upon the intellectual development of children, their cognitive development cannot be separated from their social, emotional and physical development. Experiences that promote their optimal growth toward adulthood must be considered. We must also ask how experiences with microcomputers contribute to or hinder this total development.

The designing of an appropriate instructional program for computer literacy requires the analysis of many elements, not the least of which is the nature of children themselves. In reviewing the nature of young children, we are referring to children ages three to eight, who are typically in preschool through primary grades. We suggest that there are differences among these children who, in Piagetian terms, develop along a continuum from preoperational thinking to the stage of concrete operations. We believe that children must reach the stage of concrete operations before they are ready to work with microcomputers. This is not to say that a preoperational child would be harmed by contact with a computer. However, experiences with a computer should never replace their experiences with real events and objects.

Children who have arrived at the concrete operational stage (usually around the age of seven) reason logically as long as they have concrete referents. They are able to think flexibly and to conserve the attributes of objects (for example, number and volume) when the objects are present and available for their manipulation.

Children who are in the preoperational stage, however, are inflexible in their thinking. They are, for example, unable to reason backward as well as forward and therefore cannot deal logically with equivalencies ("reversibility"); they are unable to hold two thoughts at the same time, such as is required for classification tasks and dealing with coordinate systems ("conservation"). The reasoning of preoperational children is dominated by their perceptions of the moment.

What, then, should preoperational children be doing? First of all, they *are* and *need to be* active. They spontaneously and repetitively practice physical skills in order to gain control over their bodies and their environments. Such activities serve to increase children's sense of competence and confidence. When using microcomputers, children may be actively engaged but their physical participation is limited to eye-hand coordination. Although there is opportunity for emotional and intellectual involvement, there is no chance for the mobility so essential to young children.

Real life situations provide settings in which the learning of even the youngest of the children is a natural outgrowth of their interactions with the people and things in their life-space. These functional learnings occur through problem-solving processes from which they receive feedback. The child who is building with blocks learns basic physical principles when a poorly balanced tower crashes to the floor. The child stacking pots and pans in the kitchen learns about graduated sizes. The child who kicks another and gets kicked in return quickly learns that kicking hurts. These learnings, so essential to functioning in the world, may not be replicated with a keyboard and CRT.

Preoperational children learn from real life situations, not only because of their functional qualities, but because they may offer open-ended opportunities for experimentation. Through experimentation in safe circumstances, children themselves may test their ideas and refine their understandings. For example, the child playing in sand and water develops concepts of volume, as well as permanence and change. The child "messing about" with fingerpaint learns to compose and recompose his/her thoughts and feelings in a visual medium. The child who dumps out a button box may try grouping the buttons according to their attributes such as color, size, shape, texture, number of holes, etc. There is no one right way in these experimentations.

Currently Logo, a computer language, provides children with numerous opportunities to experiment; however, the child must learn a precise set of commands before such experimentation becomes possible. Most of the popular preschool programs, such as those produced by the Learning Company (e.g. JUGGLES' RAINBOW and BUMBLE GAMES) require one right answer, and frequently necessitate the presence of an adult to read instructions.

In his forward to *Mindstorms*, Seymour Papert attributes his own personal experience *with* and *love of* gears to his ability to make cause-and-effect chains by turning wheels in his head. We believe all children need absorbing experiences with real objects that they can manipulate (just as Papert did his gears) in order to make the connection between their sensorimotor schemata and abstract concepts. Without experiencing such objects as gears, buttons, tinker toys or Lego blocks, children do not possess the foundation for the abstractions the computer presents to them.

Such activity is characterized by manipulating natural and manufactured three-dimensional materials in everyday situations. The microcomputer, on the other hand, limits the parameters of children's activity to a single viewpoint with a focus toward a keyboard and a two-dimensional, vertical surface. For example, in the software program, JUGGLES' RAINBOW, children are expected to learn the concepts and vocabulary of "above/below" and "right/left" in a functional vacuum. This may be contrasted with real-life circumstances. We know, for example, that a child climbing on a jungle gym has learned the concepts of "above" and "below" when we hear the chant: "I'm above you; . . . you're below me!" We also know that the child understands one-to-one correspondence when, on distributing graham crackers to classmates, we hear, "One for you, and one for you, and one for you . . ." We know the child is beginning to take the viewpoint of another when after a friend falls down, we hear, "Are you o.k.?"

Language acquisition is one of the major areas of learning for young children. Good models in meaningful situations are essential. At the present time, most schools are not equipped with microcomputers with voice synthesization capabilities to facilitate children's language development.[1] Children learn their language through social interaction.

> ***"The child who is building with blocks learns basic physical principles when a poorly balanced tower crashes to the floor. The child stacking pots and pans in the kitchen learns about graduated sizes. The child who kicks another and gets kicked in return quickly learns that kicking hurts. These learnings, so essential to functioning in the world, may not be replicated with a keyboard and CRT."***

In a shrinking world in which there is instant communication but limited understanding, it is essential for children to begin to develop attitudes and feelings of concern for others. These requisites to good human relationships only grow within a social context. Such relationships involve subtle clues in the behaviors of the people around them. Children, however, are able to discern through their experiences the meanings of indirect as well as direct messages in these social contexts. Although children may work together with a microcomputer, they often work in isolation. Computers provide them with unambiguous feedback and little opportunity to read significance into indirect but critical communications from others. Communications received by children at a microcomputer keyboard are direct messages circumscribed by the purposes of the programmer. Understanding these messages often requires a level of abstraction only available to a child who has arrived at the stage of concrete operations when s/he can deal with more than one factor simultaneously.

In order for children to move to the stage of concrete operations they must: actively manipulate three-dimensional objects in the physical world; discover a multitude of cause-and-effect relationships through experimentation; and develop concepts and language through involvement with others. With such experiences children begin to decenter and to recognize that the world may be viewed and represented in various ways. Young children sitting at microcomputers are being short-changed in terms of fulfilling those needs which are fundamental to learning and to their optimal development. Children at microcomputers are not using all the senses so essential to understanding the world in which they will be living. They are being limited to dealing with two-dimensional abstractions of real objects that are represented on a screen. Such representation requires a level of thinking most appropriate for children who have already had adequate early experiences with the real world.

Children spontaneously engage in the kind of activity which utilize all the senses. Of course, we are describing *play*. In play children not only employ all their senses but through it they test, refine and affirm their understanding of reality at any given point in time. Play helps them assimilate, comprehend and master their experiences.

Young children should not be rushed through these primary activities. There is a time for everything and everything takes its own time if children are to gain the maximum benefit that the experiences may provide. A child on a tricycle has more opportunities to gain a sense of control than a child on a bicycle with training wheels. In his book, *The Hurried Child: Growing Up Too Fast Too Soon*, David Elkind says, ". . . it is important to see childhood as a stage of life, not just as the anteroom to life. Hurrying children into adulthood violates the sanctity of life by giving one period priority over another. But if we really value human life, we will value each period equally and give unto each stage of life what is appropriate to that stage (1981), p. 199)."

As children move from preoperational thinking toward concrete operations, they still need to experience direct involvement with real things and with people. They are, however, ready to begin to manipulate symbolic references to real objects that have been a part of these experiences. Good beginning programs for reading and mathematics include knowledge of symbol systems, understanding of the concepts represented by symbols, and practice in manipulating them. The microcomputer has potential for providing the necessary practice in symbolic manipulation. For example, as children use Logo they must utilize symbols to achieve their objectives. However, within the constraints of the Logo language, they are free to design their own procedures. As a consequence, they are not only in command of the microcomputer but they are also in charge of their own learning. They are free to be as playful as they like.

Effective computer assisted learning (CAL) programs for young children may provide practice in using symbols; they may also employ intriguing graphics, varied formats, interesting sound effects, and immediate, nonjudgmental feedback. Typical texts and workbooks do not offer these additional significant dimensions for learning. Furthermore, good CAL programs may include tutorial experiences for children who do not understand a concept when it is first presented.

In designing programs for children in the stage of concrete operations, one must recognize the necessity of coordinating computer opportunities with noncomputer curricular experiences. Both computer literacy and CAL activities should enhance children's understandings of the world in which they live and learn. In order to accomplish this objective, one must observe children's activities and discern their meanings. Only when curricular planning is based on the developmental needs of children as well as on their own unique experiences, can one hope to provide appropriate next steps for learning.

It is the school's responsibility to provide the foundational experiences which are prerequisites to using the computer. Beyond that, there are cost-effective electronic toys such as BIG TRAK,™ which may serve as an introduction to computer programming.

> ***"Parents must be helped to differentiate between their own needs regarding their children and their children's developmental needs. Schools must help all parents understand the nature of human development and the importance of rich varied experiences for young children."***

If schools are not to foster an intellectual elite, *all* children must have access to the opportunities to learn programming skills in addition to traditional CAL experiences. If literacy is universally desirable, attention must be given to matters of equity. Without such attention, children who do not have access to computers in their homes will be truly disadvantaged.

Parents must be helped to differentiate between their own needs regarding their children and their children's developmental needs. Schools must help all parents understand the nature of human development and the importance of rich varied experiences for young children. Such experiences must include: lots of talk about the events of everyday life; opportunity to manipulate ordinary things such as sand and water; abundant lap-reading, tree-climbing and general frolicking; and encouragement of symbolic play.

Schools need to help parents understand that these varied experiences provide children with the foundation for utilizing symbols. The ability to control symbols is basic to effective learning with microcomputers. Although microcomputers are appropriate tools for young children who are at the stage of concrete operations in their development, much research is needed. Some questions which should be addressed are:

1. What are appropriate sequences for the achievement of computer literacy by children?
2. To what extent are young children capable of the linear, sequential thinking required by programming?
3. To what extent do children gain a sense of control by designing a Logo procedure?
4. Does using Logo enhance children's problem-solving and/or creative abilities?
5. Does the precision with which one must use a microcomputer help children become more reflective and analytical?
6. Does the experience of "losing a program" cause children undue frustration, or does it help them develop coping skills?
7. To what extent are "good" CAL programs, which provide feedback, branching and involvement, superior to "good" texts and workbooks in terms of more effective learning?
8. Is the current keyboard system (QWERTY) a hindrance to children's learning?
9. Do word processing programs improve children's writing skills?
10. Does the microcomputer enhance the development of children with special needs more effectively than current procedures?
11. What do all teachers need to know for effective computer instruction to occur?
12. What are the most effective ways for teachers to gain this requisite knowledge?
13. Are there adverse effects of focusing on a monitor for prolonged periods? How long should children interact with a microcomputer at one sitting?

Until we answer the above questions, we must be cautious in stating the advantages of having young children use microcomputers.

We know: there are identifiable stages of development; that abstractions are built upon the understanding of realities; and that childhood is real and should be honored. We also know that we must prepare children for the reality of living in the information age and that microcomputers are part of that reality. This is best accomplished by providing young children with concrete experiences—playing with Lego blocks *before* Logo procedures.

References

Elkind, David. *The Hurried Child: Growing Up Too Fast Too Soon.* Reading, MA: Addison-Wesley Publishing Co., 1981.

Learning Company, 4370 Alpine Road, Portola Valley, CA 94025.

Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas.* New York: Basic Books, 1980.

INVITATION TO RESEARCHERS

I am eager to identify anyone engaged in research on the use of computers with children eight years old and younger. I am interested in completed studies, work in progress and work planned. At this time I am not interested in demonstration or instructional projects unless they have a research component.

If you are engaged in research or know someone who is, please send the following information by July 1, 1983:

a) Name and address of the investigator
b) Focus of the research study[1]
c) Anticipated completion date

to:

Shirley Hill
Division of Teacher Education
California State University, Fullerton
Fullerton, CA 92634

---

[1] It should be recognized that microcomputers may play a significant role in the development of language handicapped children. Laura Meyers (Speech Communications Research Laboratory, Inc., 3500 South Figueroa St., Suite 202, Los Angeles, CA 90007) indicates that the microcomputer can be used to help young non-speaking handicapped children communicate. END■

Microworlds

by
Glen Bull

Editor's Note: Glen Bull is a professor at the University of Virginia, and has taught Logo courses at the graduate and undergraduate level.

© 1982 Posy Publications, The National Logo Exchange. All rights reserved. Reprinted with permission.

DESIGN OF A LANGUAGE

Consider the following:

- An educator suggests that a subset of BASIC (eight-command BASIC) may be a more effective introduction to programming for children than Logo.

- A salesman in a computer store tells a customer that a version of PILOT with turtle graphics is "Logo."

- A graphics language with limited test-handling capabilities is described as a new implementation of "Logo" in an article in a national publication.

The three occurrences described above have a common element in that they all betray fundamental misconceptions of the true nature and intent of Logo:

- Logo is better suited to educational use than BASIC because of the structure of the language, rather than because there are fewer commands to learn.

- Logo is more than turtle graphics. Turtle graphics is a highly visible feature of Logo which has influenced development of graphics capabilities in a number of other languages, including dialects of Pascal, PILOT and Smalltalk. Even so, powerful operators for manipulation of language are also an integral part of Logo.

Papert argues that the failure of educators to consider the design of educational languages will inevitably result in inappropriate structures, by default. The danger seen is not that the wrong choices will be made. Rather, the problem lies in the shaping of educational philosophies through lack of awareness that there are choices to be made.

The structure and characteristics of Logo are not mystical, ineffable qualities beyond discussion. They are straightforward design specifications subject to examination. To do otherwise is to act from blind faith or religious fervor.

The most important characteristics which might be desired in a language intended for use with children can be summed up in three words: interactive, procedural and extensible. BASIC was the first language designed to teach programming which was used interactively on a widespread basis. The importance of this advance would be difficult to overestimate. When centralized computing resources are expensive and scarce, they can be optimized by making people wait in a queue for processing time. Under those conditions, it may be minutes or hours before the results of a program are seen.

On the other hand, direct access to the computer and immediate results undeniably have a strong effect on learning. This philosophy suggests that a person's time is more valuable than a computer's. This may seem obvious now. At the time BASIC was developed, it was a suspect, controversial idea. It would be over a decade before the term "personal computer" was needed, in the sense of a computer inexpensive enough for exclusive use by a single person.

The widespread use of BASIC made it evident that it was highly desirable for non-professional users of computers to employ the computer in an interactive fashion. After a time, it also became apparent that BASIC was not the ultimate, perfect language. In fact, computer scientists concluded that the structure of languages such as BASIC fostered development of poor programming habits and error-ridden programs.

There are a number of reasons this is so. An important one is the fact that BASIC is not procedural, in the true sense of the word. Its closest approximation to a procedure, the subroutine, falls short in several respects. In a programming language, a procedure is an independent description of a process which performs an action. Once developed, this description can be treated as a black box by the remainder of the program. The mechanics of how the goal is accomplished inside the black box can be disregarded.

The best-known procedural language developed for teaching programming is Pascal. It has almost universally been adopted for teaching programming by computer science departments. Pascal was developed for college students. Hence, a considerable knowledge of computer science is required to write even a short program in Pascal. Also, Pascal programs cannot be run as soon as they are typed, but must be compiled first. The benefit of Pascal is that it does encourage good programming practice, even when the user moves on to other languages.

Logo is both interactive and procedural. A procedure can be run as soon as it is typed. A design goal of Logo was an educational language with no threshold and no ceiling. In a certain sense this goal has been met, since Logo has been successfully used by children in elementary school and by physics students at MIT. In passing, it might be mentioned that the common impression that Logo is only of interest to children is incorrect.

ilities for developing routines in assembly
well as list processing capabilities found in
es.

procedure is invoked by naming. As a con-
e language is extensible. This means that
develop new commands in Logo and, in
a new language tailored to a particular
ication. This capability is only available in
languages, such as Forth. For example,
not provide a FOR-TO control structure
se found in BASIC and Pascal. The philo-
phasis of Logo seems to be on recursive
iterative control structures. However, a
mand can easily be developed in Logo, if
ed for this structure.

ent implementations Logo is, unsurpris-
erfect language. The intended application
e heavily influences its efficiency. Logo is
ly fast. Because it is a sophisticated lan-
uires more memory than languages such as
lassroom environment, speed of execution
use of memory are usually not the most
derations.

of silicon computer chips will probably fall
their capabilities will increase. As this oc-
ons of speed and memory will become less
However, language characteristics which
of technological limitations will continue to
ted unless thought is given to desirable
ample, usually the absence of procedures
esursively call themselves is the result of
speed limitations, rather than a conscious
a language would be improved by the absence of this capability. Therefore, if these features are desirable, a continuing review is necessary to ensure that the omission of the feature does not continue after the technologic constraints are removed.

Logo was developed over a number of years in an environment in which cost was not a constraining factor. The only stated limitation was usability of a characteristic in a learning environment. This process was begun over a decade ago. Now some of the benefits of this effort are being reaped as many of these characteristics are becoming possible in microcomputer-based implementations.

Three important aspects of Logo are embodied in the fact that it is interactive, procedural and extensible. This by no means exhausts the list of capabilities which define Logo. Other aspects such as encouragement of meaningful procedure and variable names by removing restrictions on their length, the presence of a resident screen editor, and effective means of passing parameters between procedures all contribute to the underlying quality of Logo.

Some components of Logo can be altered or deleted without seriously degrading its fundamental structure. Some implementations of Logo have arrays, sprites and floating-point arithmetic, while others do not. Depending on the intended use, these omissions may not seriously cripple the language. However, beyond a point it becomes questionable whether a language can seriously be described as "Logo," given the original intent of the language. Effective utilization of Logo will ultimately depend on teachers who understand its characteristics and the reasons for their inclusion in the language. END■

November 1982 The Computing Teacher

Creating a Logo Environment

by
Tim Riordon
Alpine Educational Computing Services
3375 University Street
Eugene, OR 97405

Author's Note: Perhaps you are a curriculum specialist, school principal or classroom teacher interested in introducing Logo in your district, school or classroom. You've done some reading and have a grasp of the philosophy. Now, however, it's time to implement your ideas. How do you actually create a Logo environment? This article bridges the gap between theory and the beginning of actual practice.

The training in Logo that many teachers are receiving focuses upon learning the Logo language. In four hours, a day, or perhaps a week, teachers begin learning to use Logo commands to write procedures to do wonderful things—graphic designs, music, robot control, random poetry, arcade-like games, etc. This training is necessary but not sufficient for most teachers to use Logo as it was intended. Most teachers need not only to learn the langauge but also to learn how to *implement* Logo—how to create a Logo environment in the classroom.

This article examines five questions:

- What is a Logo environment?
- What are its attributes?
- What things are conspicuously absent in a Logo environment?
- What are the reasons for creating such an environment?
- How does one begin?

Ideally, a Logo environment would have a number of computers with many students doing Logo at the same time. However, such environments will not occur in most schools until school districts begin to pool computer resources and lend individual teachers sets of Logo machines. Therefore, this article is written for the classroom teacher who has only one computer and wishes to create a Logo environment.

What Is a Logo Environment?

It is the entire context, made possible and managed by the teacher, in which students work with Logo. It is more than a computer learning station. It includes psychological as well as physical space—how students feel, how students and adults interact.

What Are the Attributes of a Logo Environment?

What would an observer see, hear and feel? Suppose an omnipresent observer—perhaps a principal doing an evaluation—were to spend a week in a classroom with 25 students, one computer, and a Logo environment. What would s/he observe?

It is Monday morning. Teacher and students have not yet arrived. In the front corner of the room, not far from the teacher's desk, is a table with two chairs, a computer with disk drive, a large television monitor and supplies. Supplies include masking tape, rulers, graph paper, pencils, felt tip pens and drawing paper. There is a box with three disks in it labeled "LOGO STORAGE DISKS." A small weekly schedule shows each day divided into 20-minute periods. The schedule is nearly filled with students' names. The words "Whole Class" appear four times on the weekly schedule. Each student uses the computer 3 times per week while the rest of the class continues with other instructional activities. On one wall is a chart with student names down one side and several projects listed across the top of the chart. Many students have checked off projects after their names. Above the chart is the sign "TRY TWO OF THESE PROJECTS THIS WEEK," a direction which seems to invite students without commanding or threatening. On the other wall are some computer printouts of student designs signed by the student programmers. There are also drawings done in felt tip pen—some by students, some by the teacher. These drawings are on a section of the wall labeled CAN YOU MAKE THE COMPUTER DO THIS? This is a problem-solving environment. There appears to be a sharing of authority since both students and the teacher are allowed to contribute to the collection of problems. On the floor are many designs made with masking tape. What purpose do these serve?

The teacher arrives and removes some of the drawings from the walls. Four new problems are posted. Three of them are designs and one is a question: "CAN YOU WRITE A LOGO PROCEDURE THAT WILL PUT YOUR NAME INTO A CODE?" The students are evidently using Logo for more than graphics exploration.

School begins and students, sometimes alone and sometimes with a partner, go to the computer at their assigned times. For the most part, transitions are smooth. Kids are eager to get to the computer, reluctant to quit and store their unfinished business. Once the teacher has to intervene: "Mary, you really must finish now. It's Kevin's turn. There's some open time at lunch recess I'll let you use if you wish." During the time each student works at the computer, the teacher finds a few minutes to watch and sometimes, seeing a student encounter an unexpected result, asks quietly, "Do you have a theory about what caused that?" Most interaction with students about their Logo work, however, occurs at another time. Occasionally, the teacher will direct something to the entire class, including the student working at the computer. "Kevin, I need to interrupt you so you can hear an announcement for the whole class. . ."

Several times throughout the day students are seen leaving the computer, going to another student in the class, interacting and then returning to the computer. Sometimes both students will go to the computer for a short time. Then the student who was doing seatwork will return to his/her seat and continue working. Listening to these interactions, one hears, "I can't remember how to make the cursor go down"; "How do you make the turtle appear randomly at different places on the screen?"; "SETXY isn't working right"; "How do you erase a line you don't want?" The explanation for this is that each week two students volunteer to be LOGO EXPERTS. Any student experiencing difficulty while working at the computer has permission to seek the help of one of these experts. There is one boy and one girl, and it should be noted that the experts are not always the most academically advanced students. The teacher is sharing authority and at the same time greatly reducing the number of interruptions which would otherwise occur.

About twenty minutes before lunch recess, the teacher ends a language arts lesson and walks over to the computer. The schedule says "Whole Class." A few students sitting in the back move forward in order to see better. After asking for volunteers to demonstrate some of their work, two students show a procedure that draws a submarine in water. An airplane is then drawn in the sky. A bomb appears beneath the plane and slowly falls toward the sub, causing an explosion when it hits. This demonstration is followed by a student who demonstrates a procedure that draws Garfield the Cat at random places on the screen. Whenever a student demonstrates a procedure, there follows a discussion, shaped by a questioning strategy composed of four questions:

1) ***Do any class members have questions about the procedure we've watched?*** Several students usually want to know how it is done, so the demonstrators show the inner workings of their procedures. Occasionally, a value-related question arises when a student refuses to show how a procedure works. "I don't want to show it—it's a secret." This leads to a brief discussion about who owns knowledge, and about copyrights and patents. (Usually this motivates one or two other students to discover the secret which is later gleefully published!)

The second and third questions are addressed to the student who demonstrated the program:

2) ***What do you plan to do with your procedure now?*** Sometimes the answer is, "Nothing—I'm done with it." Other times elaborate plans are given. Frequently, the student will respond in a sketchy, tentative manner as in these examples: "We'd like to have it so that a person can aim and fire the bomb"; "I'd like to have some music go with the drawing"; "I'm going to try to figure out how to make each cat a different color." The tentative nature of these responses tells the teacher that the student has encountered a new problem to be solved but does not yet know the solution. Such responses are always followed up by another question:

3) ***Do you have some ideas to try out or do you need some hints about how to do it?*** The teacher is seen making notes for students wanting help. The notes read like this:

Ben—give him music routine.
Sarah—have Ann show SETXY and RANDOM
Thane—hint about multiple colors
Keri—tape & walk rainbow starting from center
Raol—no help needed

Several things about this environment are becoming evident. Students decide for themselves whether they need help solving a problem and in so doing they learn to reflect upon, and be responsible for their own learning. Some students who want help are given the complete answer to their problem, sometimes by the teacher and sometimes by another student—authority is distributed. Students often are given hints but not complete answers; when guided discovery is used, students frequently experience the joy of finding the solution by themselves.

The final question raised after each demonstration leads to a short brainstorming session by the class:

4) ***What are some things which could be done to change this program?*** Many fascinating Logo projects result from ideas contributed from several class members. The teacher also contributes ideas—although usually after ideas have emerged from students.

The whole class responds to work demonstrated by class members two other times during the week. And on one other occasion, the teacher demonstrates a new Logo programming idea to the entire class. (Earlier in the year when Logo was first being introduced, teacher demonstrations outnumbered student demonstrations. Now the balance has shifted.)

Throughout the day the teacher is seen consulting a list with eight student names. When a few minutes are available, the teacher looks at the list and selects one student for an impromptu 5-minute conference. Tomorrow another eight names will be added to the list. The first question in these conferences always seems to be, "What are you doing with Logo this week?" Some students have elaborate plans for games or procedures. Other students, exhibiting a very different manner of working, explain that they are "messing around" with one procedure. By "messing around" they apparently mean to work with one procedure, make a change in it, run it, make another change, run it again, etc. Evidently, the teacher does not try to coerce students into adopting another method. "Messing around" with Logo procedures is valued. The teacher is sensitive to and accepting of different intellectual styles. The second set of questions is usually, "Have you had any unexpected results? Do you have a theory about what caused this? What did you try in attempting to solve the problem?" The teacher is using questions to help students focus upon their own theorizing, and is promoting learning by having students theorize, test and revise, instead of always having them learn by direct instruction.

"Often the teacher will tell the student to use the 'masking tape strategy.' Using this strategy, a problematic design is examined and then taped on the floor. The student plays turtle and walks through the procedure. Using Logo language, the student describes everything s/he is doing to walk the design. Sometimes the teacher or another student copies down each statement. The result is usually the Logo procedure needed to create the design in question.

The central question in these short conferences always seems to be, "Is there anything you are having trouble with on your current Logo project?" If the answer is yes, the student is asked to draw what happens and if s/he has any ideas about what might be causing the problem. Frequently, the problem is complex enough that the student and teacher agree that they need to watch what happens on the computer. Somehow a few minutes at recess, lunch or after school are found so that the problem can be demonstrated. Sometimes the teacher provides the student with an answer. Other times the teacher will give hints and encourage the student to try another approach to solving the problem. Often the teacher will tell the student to use the "masking tape strategy." Using this strategy, a problematic design is examined and then taped on the floor. The student plays turtle and walks through the procedure. Using Logo language, the student describes everything s/he is doing to walk the design. Sometimes the teacher or another student copies down each statement. The result is usually the Logo procedure needed to create the design in question.

Sometimes students will say that they are done with a project or are bored with it. With some students, the teacher's resources for project ideas are offered as possibilities: *Logo for the Apple II* by Abelson, and *Final Report Brookline Logo Project, Parts II and III* by the MIT Logo Group. With other students, the teacher encourages and occasionally directs that something specific be done—usually a project that the teacher believes will open up new Logo vistas for that particular student. "Bill, I'd like you to do a little project that makes the computer draw umbrellas of increasing size all over the screen. Do you have an idea about how you would do that?" (Bill understands how to use variables and recursion but has not used the two ideas together; the teacher makes sure that Bill has some idea about how to begin so he will not experience great frustration.) "Katie, would you rewrite this procedure that always draws 9-pointed stars the same size? Make it so that you can input both how big and what color the 9-pointed star is. Do you have some ideas about where you would change this procedure?" "Matthew, I think that you will be able to solve your problem if you try the project on the wall that asks you to use a circle to make a slinky design. If you do that project and still can't figure out how to do what you want in your current project, come on back and see me."

After watching and listening to these 5-minute interactions throughout the course of a week, several conclusions can be drawn. The teacher:

- is sensitive to whether or not students are headed for frustration.
- has in mind a sequence of Logo programming concepts and a large repertoire of Logo project ideas.
- often considers his/her teaching role not as a repository of answers but as a midwife helping students discover answers by theorizing about problems and unexpected results.

Using all of this knowledge, the teacher exercises the art of teaching in quickly discerning how to respond to each student as an individual. Some need an answer and some just a hint; some need a practice project; and many need nothing but more computer time.

A Logo environment has many of the attributes of a democratic classroom. Authority is distributed; sharing and cooperation are promoted; students look to their classmates as legitimate sources of information; because students make project choices, variety rather than uniformity is the norm; rewards are intrinsic; differences in working style are valued. There is a sense of shared learning—students may hear their teacher respond, "I'm not sure how to make the computer do that. I'll work on it at home tonight."

What Features Are Conspicuously Absent from a Logo Environment?

There is no evaluation of student learning prepared by the teacher and given to students or parents. There are very few required projects which students are expected to do. Many of those that are required are selected specifically for an individual student rather than for the whole class. While there is no formal evaluation of a child's Logo learning, parents are kept abreast of what their kids do, sometimes through a parent night, where students demonstrate their programs. Logo lends itself to class projects, booklets and calendars with Logo graphic designs by each class member. It should also be noted that Logo excites students enough to go home and volunteer information to parents on "what we did in school today."

Why Should Such an Environment Be Set Up in a Classroom?

Why should classroom instructional time be used by students in an activity which is not formally evaluated?

> ***A Logo environment is a mathematics-rich environment, a context including not only a computer and walls and floors with project ideas but also how students and the teacher in the classroom interact with each other. In an environment where mathematical objects and ideas are joyfully shared, played with, discussed and encouraged, mathematical intuitions and language will be learned without excessive formal instruction."***

The most complete answer to these questions is to be found in Seymour Papert's book, *Mindstorms: Children, Computers and Powerful Ideas*. Logo was created because of a fundamental belief that children can learn many important things without formal instruction. Just as children learn verbal language without formal instruction—a vocabulary of thousands of words and a set of complex syntactical rules for putting words together meaningfully—so too the language of mathematics can be learned. No matter what one's theory about how language is learned, it is a fact that children learn language when they respond to, and are responded to, by a language-rich environment. A Logo environment is a mathematics-rich environment, a context including not only a computer and walls and floors with project ideas but also how students and the teacher in the classroom interact with each other. In an environment where mathematical objects and ideas are joyfully shared, played with, discussed and encouraged, mathematical intuitions and language will be learned without excessive formal instruction. What lies behind all of this is a Piagetian view of children who learn because they naturally make and revise theories about things they are interested in.

How Does One Begin Creating a Logo Environment?

Once you begin learning Logo yourself, introduce it in your classroom. This article was written with self-contained classroom teachers' in mind. In middle schools, high schools and other situations that are not self-contained, teachers can invent other ways of organizing. Get Logo going as a peripheral activity; that is, let your regular curriculum proceed but allow individual students to miss parts of it. Students who miss instruction can be treated like students who are absent. With older students, it is possible to strike a bargain—a classroom group agreement. A student working on the computer will be responsible for checking with a friend about what s/he has missed.

In the beginning you will need to demonstrate a number of things to the whole class. The question of scope and sequence is problematic. Logo inventors fear that if a scope and sequence is published it will invite the belief that students should be accountable for learning Logo programming concepts. This will inevitably lead to evaluation of student learning. This will lead to Logo being a joyless, unnatural learning activity. Logo was not meant to be taught, and the kind of learning that occurs was not meant to be evaluated like other school learning!

With the above warning, I present a sequence I used recently:

- Moving the turtle around the screen and the REPEAT command
- Saving pictures on disks
- Creating procedures—TO command
- Saving procedures on disks
- Editing procedures—moving cursor around in the editor
- Problem-solving strategy—tape it, walk it, turtle talk it
- Placing designs on screen—RANDOM and SETXY primitives
- Using variables
- Doing and printing arithmetic
- Using procedures in other procedures
- Analyzing designs—looking for repeated patterns
- Simple recursion (tail recursion)
- Taming recursion TEST, IFTRUE, IFFALSE, IF-THEN STOP primitive
- Recursion with variables
- Analyzing designs—repeated but slightly changing patterns
- Dynaturtle
- Making a "hot" keyboard (individual keys make things happen)

- Music procedures
- Printing numbers, words and lists—WORD, SENTENCE primitives
- Superprocedures and subprocedures
- Planning a game
- List processing commands—FIRST, BUTFIRST, LAST, BUTLAST primitives

This sequence is very flexible. For example, printing numbers, words and lists, and writing music could be taught to students in the first few days.

Once you have students starting to create designs, begin the "Whole Class" demonstrations where students show their work. Try the questioning strategy mentioned in the article. As you begin responding to students who say, "I can't . . .", try to get them to theorize about what causes the unexpected to occur. Show them how to use the TRACE command. Make them use tape and walk the design they are aiming for. Try seeing yourself as a facilitator and not as an answer machine.

Don't let the readings suggested below prevent you from starting. It is in the spirit of Logo that you should begin the project of creating a Logo environment with the idea that you will try something, debug it and try again.

References

Abelson, Hal. *Logo for the Apple II.* New York: BYTE/McGraw Hill, 1982.

Adams, James L. *Conceptual Blockbusting.* Stanford, CA: Stanford Alumni Association, 1974.

BYTE Magazine. Entire issue, August, 1982.

MIT Logo Group. *Parts II & III Final Report Brookline Logo Project.* Cambridge, MA: Artificial Intelligence Publications, Mass. Institute of Technology, 1979.

Papert, Seymour. *Mindstorms: Children, Computers and Powerful Ideas.* New York: Basic Books, 1980.

Suchman, Richard. *Developing Inquiry.* Chicago: Science Research Associates, 1966.

END■

September 1983 The Computing Teacher

Turtle Fever

by
Rick Billstein

Just as Pac Man™ fever became the rage of elementary and secondary students around the nation, it appears that a new fever has recently struck the educational community. This fever is not dominated by a circular character with a large mouth which devours power capsules when moved around the video screen. Instead, the principal character is a plain looking triangular figure which is simply known as a "turtle." The turtle is not the hero(ine) in a video game, but rather the display creature that resides on the computer screen in the "turtle graphics" portion of Logo.

Logo is a computer language developed over a 12-year period by Seymour Papert and the staff of the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology (MIT). Papert refers to the turtle as "an object to think with." Using very simple commands called "primitives," students can learn to control the turtle by moving it forward, back, right or left. The turtle can leave tracks on the screen or it can move without leaving tracks.

Pac Man™ involves problem solving under time pressure in that points must be scored and enemy goblins must be evaded or destroyed. However, the problem solving involved in "turtle geometry" has no time pressures—that is, if you don't count the pressure from other students waiting to get on the machine.

Logo allows students to pace themselves. The machine responds no matter how fast or how slowly the students work. Students create and discover many things on their own or through working with their classmates. The satisfaction and the enthusiasm that these students have about the computer and about themselves is something that I don't usually see when students finish a course in BASIC. I have felt their excitement when teaching seven Logo classes for teachers and undergraduates and during my observations of elementary students in classes and computer camps.

I'm not alone in my enthusiasm. All one has to do to see evidence of turtle fever is to pick up any computer education journal. There are articles on Logo in most of them and several now have columns on Logo. Some new journals are devoted almost entirely to Logo. Logo user groups are forming all across the country. Sessions on Logo at computer conferences and mathematics conferences are filled to capacity. Universities and colleges have trouble handling the demand for Logo classes. Since I started teaching Logo classes two years ago, the number of requests I receive for materials and information has continually increased. I worry that when I tell people about Logo and about the impact I think it will have on education, they will think, "Wait a minute, aren't you the same guy that told us the metric system was coming?"

I would have trouble defending my position statistically, but for now the feelings I have are enough for me. Logo is good enough to survive in spite of what mistakes we educators may make. I also believe that educators will do more good with Logo and make fewer mistakes if they avoid some misconceptions, are aware of some implementation problems, and have a good grasp of uses of Logo.

More Than Graphics

Logo is powerful because it is more than just turtle graphics. Turtle graphics are very important in the language but the list processing capability of many versions of Logo is also important. Lists are all-purpose data structures that are particularly well-suited for artificial intelligence applications. (For an example of a use of the list processing capability of Logo, see "List Processing in Logo" by Tony Stavely, *The Computing Teacher,* Vol. 10, No. 4.) Whether or not teachers should become proficient with the list processing capability of Logo is another topic, but at least they should be able to follow simple list manipulation routines. My experience is that teachers become attracted to Logo because of turtle graphics and do very little else in the beginning. After a time, many explore the other features of Logo and become very enthused about them. This is true even though list processing is harder to learn than turtle graphics. (This problem has prompted Bob Tinker in an article called "Logo's Limits: Or Which Language Should We Teach?" to begin talking about the need for a "beyond-Logo" language.

Not Just for Kids

Logo is usually advertised as a computer language for children. Logo is not just for kids. It is a powerful, general-purpose programming language that can be used with students at all levels. For example, Professor Andrea di Sessa of MIT uses Logo to teach physics to undergraduate students. He created a "dynaturtle," a dynamic turtle which obeys Newton's Laws of Motion. Working with the dynaturtle helps students better understand how forces affect the motion of objects. The depth of Logo and chess are alike. Elementary students can learn the basic tactics and pleasures of chess, but the grandmasters are still striving to learn more about the game. The designers of Logo refer to it as "a language with no threshold and no ceiling." For an excellent article which discusses many of the powerful features of Logo, see "Why Logo" by Brian Harvey in the August, 1982 issue of *BYTE* magazine.

Logos, Not Logo

Not all products sold with the Logo name are alike. Different companies have built different capabilities into their products. For example, some products will do only the turtle graphics portion of Logo and no list processing. Some turtle graphics accept only integer inputs while others allow floating point (decimal) entries. Each version of Logo will run on only one brand of computer.

Some, but not all, dialects have "sprite" capability. A sprite can be thought of as an invisible character to which the user can assign attributes. Sprites can carry a shape attribute that the computer already knows or a shape the user designs. In addition, sprites are given color, speed, direction and position. Students can use sprites to form animated graphics. Some Logos such as Texas Instruments Logo have sprite capability built into their product. On other systems, a sprite graphics board and software must be purchased.

Because of the differences in Logos, people are finding that some interesting Logo activities that they would like to try will not work with their setup. This is not all bad since then we have a new problem-solving experience—understanding the program and translating it into the dialect understood by the user's computer. However, it can be very frustrating to spend a large amount of time typing in a program only to find that some of the symbols in the program are not compatible with the language loaded into the computer.

Implementation Difficulties

Logo cannot be all things to all people. If speed is important, users will find that the present versions of Logo are slow in comparison to BASIC. Also, the current implementations of Logo have little user workspace. For example, when students in a summer course tried to write a program that would print out all of the letters of the alphabet, they ran out of storage before they were finished. In addition, Logo has trouble handling many numerical applications. For example, when students tried to write a program to compute compound interest, they found that they could not compute daily interest because Logo could not handle numbers raised to the 365th power.

The problems just discussed are problems that can be fixed with more advances in hardware. If the advances of the last few years are indicative of the future, we can look forward to newer, more powerful versions of Logo with much greater capability.

Preparation First

Since Logo is so new, most instructors of inservice and preservice classes on Logo have relatively little background on the subject. Thus, in some situations, the power and uses of Logo may not be as evident as they could or should be. In addition, there is a shortage of resource materials, guidebooks, completed research projects, etc. I am beginning to worry if elementary teachers with a weekend workshop from an instructor with little Logo experience will be able to do an adequate job of implementing Logo into their classrooms. Logo classes must not only discuss syntax and capabilities, but also the philosophy and logistics of exposing Logo to others.

Although Logo is advertised as a language in which the users discover things on their own, *teachers must know Logo.* They must know Logo well enough to guide and encourage students. Teachers should not solve problems for the students, but they should know enough Logo to give appropriate hints before the students' frustration leads to unproductive results. Successful problem-solving experiences require problems that are interesting and challenging but not frustrating beyond students' emotional or intellectual capacity.

> ***"Although Logo is advertised as a language in which the users discover things on their own, teachers must know Logo."***

Learning problem-solving skills is a process which takes time and a reasonable environment. Students love to work on the machine and they become very engrossed in their own projects. But when students experience shortened class periods, limited machine access, and increasing amounts of course content to cover, the marvelous gains that educators expect from Logo may not be as evident as we would like. Good classroom management and administrative support are critical considerations when implementing Logo. Decisions will have to be made about how much hands-on time each student needs, whether students should work in groups and if so what size groups, and whether to allow students time on the computer at the expense of other activities taking place in the classroom.

Some Things, Not Everything

We must not expect students to learn everything on a topic just from their Logo experiences. For example, there are topics in the mathematics curriculum that cannot (or should not) be Logo driven.

For example, it would be very hard to study concepts such as surface area or volume using Logo. Children not only need to work with objects that can be drawn on a flat screen, but also with three-dimensional real objects. Instruction in this area must not be neglected. Areas where Logo is useful in mathematics include providing an environment for developing mathematical concepts such as variables; the group properties of numbers; the geometrical concepts of length, angle measure, symmetry, similarity; and coordinate systems.

The use of Logo in itself certainly aids the teaching of computer literacy. Students exercise hands-on control of a computer, learning its capabilities and limitations as they learn a structured programming language. Logo seems to be a useful tool for developing logical thinking and problem-solving skills such as the use of top-down or bottom-up strategies.

With All Deliberate Speed

I have been discussing these points about Logo with many people lately. I'm saying, "Yes, I feel that Logo is good, but let's slow down a little."

> ***"Areas where Logo is useful in mathematics include providing an environment for developing mathematical concepts such as variables; the group properties of numbers; the geometrical concepts of length, angle measure, symmetry, similarity; and coordinate systems."***

"Everyone else is doing it" is not a good enough reason to immediately run out and buy the first Logo system available. Remember that Logo implementations often do not work as nicely and easily as you read or hear about. Just as in the story about the tortoise (or in this case, turtle) and the hare, it is sometimes better to trudge along with caution rather than bound about without knowing more about a program and what we want to do with it.

First, get acquainted with Logo. Spend some time finding out how implementations differ. Try to understand the power and limitations of Logo. Consider not only what you want to do with Logo right now, but also what you may want to do with it in the future. Set realistic expectations based on your situation. Consider what hardware is available and whether you can afford it. Will purchasing cheaper hardware now really be more expensive in the long run when everything needed for expansion is considered? Who will handle the teaching, and is adequate training feasible? How easy will it be to accept a new role as a Logo teacher? Answer those questions before hardware and software appear in the classroom.

As Molly Watt has written, "As Logo teachers, our roles will cover a diverse, and perhaps uncomfortable range, including demonstrator, teacher/lecturer, teller, time structurer, problem setter, management solver, arbitrator, decision maker, challenger, helper, collaborator, process sharer, question asker, idea extender, observer, documenter, admirer, enjoyer, time providers, technician and model learner." It's great to catch turtle fever, but like Pac Man,™ let's know what we are gobbling up and where the goblins are. Let's do it right.

END■

Rick Billstein, Department of Mathematical Sciences, University of Montana, Missoula, Montana 59812.

BIBLIOGRAPHY

Harvey, Brian. "Why Logo," *BYTE*, August, 1982, pp. 163-193.

Stavely, Tony. "List Processing in Logo," *The Computing Teacher*, December 1982, pp. 43-47.

Tinker, Bob. "Logo's Limits: Or Which Language Should We Teach?", *Hands On*, Vol. 6, No. 1, pp. 1-6.

Watt, Molly. "What is Logo," *Creative Computing*, October, 1982, pp. 112-129.

Keyboarding—A Must in Tomorrow's World

by
Evelyn Kisner

Article after article has been written about using computers for educational instruction. These articles provide helpful suggestions and directions for using computers as instructional tools. Yet one very important factor in learning to use a computer efficiently has not been emphasized enough—*keyboarding!*

Just What Is Keyboarding?

Keyboarding is the term used to describe the activity of entering information on electronic equipment through the use of a typewriter-like keyboard. Keyboarding is the skill that enables an individual to use a computer terminal or microcomputer efficiently.

Who Needs Keyboarding?

Almost everyone! Projected use of keyboard terminals at all levels of activity—at home, school, work and recreation—clearly indicates that keyboarding is becoming a necessity.

The increasing impact of computers as an instructional tool emphasizes the need for all students to learn to operate an electronic keyboard, by touch, if they are to function effectively. Without keyboarding skills, students waste valuable learning time searching for the right keys.

Some institutions of higher learning are refusing computer time to students until they have acquired keyboarding skills. The reason: Computer time is too costly to be wasted while some students hunt-and-peck.

Is Organized Instruction in Keyboarding Needed?

Yes! Keyboarding is a psychomotor skill, and correct techniques are necessary for mastering such a skill. As students begin using the keyboard, they form lifelong habits where correct techniques are extremely important for efficiency. The most effective way to develop keyboarding skill is through organized instruction.

Although the two-finger, hunt-and-peck method may work for short response, such a method is not adequate when more lengthy keying is necessary. The two-finger habit requires looking back and forth from copy or instructions to the keys, searching for the right keys, then looking at the screen and back to copy. This habit is time consuming and, once formed, is difficult if not impossible to correct.

When Should Keyboarding Be Introduced?

Keyboarding should be taught prior to the grade level where students begin using computers for academic work, with a goal of mastery of the keyboard before students begin working with instructional software programs. Being able to keyboard, by touch, enables students to concentrate on instructional responses. (Suggestions for competence levels for different grade levels are provided below.)

Who Should Teach Keyboarding?

Although elementary teachers may be able to keyboard through transfer of typewriting skill, it is doubtful that any are prepared to teach keyboarding.

> ***"Keyboarding should be taught prior to the grade level where students begin using computers for academic work, with a goal of mastery of the keyboard before students begin working with software programs of instruction."***

Business teachers are currently the only group with the background preparation for teaching keyboarding. They have studied the psychology and methods for teaching psychomotor skills and have been applying those methods to teaching the typewriter keyboard.

How Should Keyboarding Instruction Be Implemented?

Some school districts preparing for extensive use of computers for instruction at the elementary level have developed workable solutions for meeting the keyboarding need.

Several options have resulted. In Omaha, Nebraska,[1] a conclusion was reached that a team-teaching approach with business and elementary teachers working together to develop instruction for keyboarding could result in the most significant gains.

The following options, with business teachers as instructors, were developed in Lincoln, Nebraska:[2]

- Elective courses in summer school with a tuition charge and parents providing transportation to a school where electric typewriters are available.
- Evening or late afternoon classes at a high school where electric typewriters are available. Parents would provide transportation.
- Taking machines and teacher to the elementary school. This would require a traveling teacher and a room of electric typewriters or microcomputers. The equipment and teacher would remain in a school four or five weeks, and then be moved to another school. With class periods of 30 minutes each day, students who wanted keyboarding skills could be accommodated.
- In some areas, smaller schools might share a traveling van equipped with furniture and electronic typewriters or microcomputers. This option utilizes a traveling teacher. One advantage of this option is that the furniture would be of suitable size and height for elementary students.

What Level of Keyboarding Efficiency Is Needed?

The objective of initial instruction in keyboarding for students at the elementary level should be mastery of the keyboard, by touch, with mastery to be achieved through:

- Correct posture at the machine.
- Proper techniques for striking keys.
- Accurate fingering of keys.
- Operation of basic machine parts.

For kindergarten through Grade 3, where students' input is generally numbers and one-word responses, achievement of 20 words a minute with accuracy could be considered adequate for responding to software instructions. Most students should be able to achieve this rate in four or five weeks.

For Grades 4-6, where input generally calls for complete sentences, the achievement goal might be set at 25 words a minute, with accuracy.

The financial and scholastic challenge of meeting elementary students' need for keyboarding skills requires support from the principal, the superintendent and the school board. The development and implementation of keyboarding instruction depends upon the cooperation of elementary teachers who have expertise in teaching elementary students and business teachers who have expertise in teaching keyboarding skills. Teamwork can result in an appropriate program for meeting the challenge.

END■

[Evelyn Kisner, Director, Business Education, Texas Education Agency, 201 E. Eleventh St., Austin, TX 78701.]

---

[1] Verda C. Rauch and Patricia B. Yanke, "Keyboarding in Kindergarten—Is It Elementary?" *Business Education Forum*, December 1982, pp. 19-22.

[2] Jane Stewart and Buford W. Jones, "Keyboarding Instructions: Elementary School Options," *Business Education Forum*, April 1983, pp. 11-12.

April 1984 The Computing Teacher

Sex Equity: Increasing Girls' Use of Computers

by
Marlaine E. Lockheed and Steven B. Frakt

As a relatively new field, computer science is free of the male preeminance associated with the traditional sciences. Women have made valuable contributions to the development and application of computers. For example, the world's first programmer was Augusta Ada Lovelace; the defense department's computer language ADA is named in honor of this 19th century pioneer. The first major computer, the 1940's ENIAC, was operated by 100 female programmers.

As the computer era leaves its infancy, however, there is mounting evidence that it is coming of age with putative sex-linked traits. Considered by the general public to be heir to a long line of home-workshop electronic gadgetry, servant of mathematical modeling, purveyor of aggressive games, and apparently fathomable only by male systems analyst cognoscenti, the computer is gaining notoriety as a preeminantly masculine technology.

> ***"By male self selection and female default, the computer center becomes defined as 'male turf'—as socially inappropriate to girls as the boys' locker room."***

The stereotype of the computer as a male machine has emerged concurrently with the tremendous growth in the use of microcomputers by children and is due in large measure to the manner in which microcomputers have been introduced, promoted and adopted in the family room and in the classroom. It is important that educators be aware of the processes that give rise to the stereotype of the "male" computer and that they take action to overcome this stereotype by encouraging the use of the microcomputer as the general purpose, gender neutral tool that it is.

A number of recent studies have indicated that, while girls and boys show similar appreciation of the significance computers have for their personal futures, boys are gaining a greater working knowledge of computers by spending time on hands-on computer-related activities. Survey after survey shows that the present generation of school-aged girls maintains a high degree of awareness of and interest in computer technology. In a statewide study of 17,861 California students, nearly three-quarters of the twelfth grade girls and two-thirds of the sixth grade girls agreed that "a knowledge of computers will help to get a better job." The 1983 Gallup Youth Survey reported that 65 percent of girls aged 13-19 planned to take computer courses in college and over half of that group thought it likely that they would have a computer-related major in college. In a recent survey of high school students enrolled in a mandatory computer science course, 80 percent of the girls (as compared to 82 percent of the boys) agreed that knowing about computers would be important for their own future. Still another study found no sex differences in sixth-grade students' self confidence regarding computers, in the students' perceptions of the utility of computers, or in their attitudes toward computers.

These same surveys, however, also show that boys have greater access to computers than do girls and use them more. One study of 400 students enrolled in a required introductory computer science course found that almost half the boys, but virtually no girls, used the computer center outside of the required class time. If the girls had not been enrolled in the required course, it is quite possible that they would not even have known about or ever used the computer center. (This latter point is supported by the findings of a major study in which more boys than girls reported having microcomputers at school.) Differences in access to computers outside of school have also been reported, with more boys having access to a computer at a friend's house, more boys learning about computers from friends and at stores, and more boys learning about computers at computer camps. All these studies indicate that if boys are gaining an edge in computer technology, it is not due to sex differences in interest toward or understanding of the relevance of computers but to sex differences in access to and use of computers.

What factors account for this difference in computer use, and what strategies can teachers adopt to equalize computer use by boys and girls?

Sex Segregation

Researchers have demonstrated repeatedly that boys and girls in grades K-8 typically lead highly sex segregated lives, and that this segregation persists into secondary school for most activities that are not directly related to dating. An example of how sex segregation affects classroom behavior is provided by the following observation of a fifthgrade science class:

For an experiment regarding light, each child was given an unexposed piece of photographic paper and was instructed to carry out a set of procedures on the paper. When these procedures were completed, each child was to go to one of two light sources in the room—a window and a lamp—and to expose his or her paper. The first child to finish was a boy; he went to the window. The second child was a girl; she looked at the boy by the window and then went to the lamp. The next two children to finish were a boy and a girl; the second boy followed the first boy to the window and the second girl followed the first girl to the lamp. The rest of the class followed suit, each child going to the "sex appropriate" light source. By the end of the experiment, the girls had used only the lamp and the boys had used only the window.

Substitute "computer center" for "window" and it becomes clear how computers—like windows—can become a male sphere of influence as a result of sex segregation. Even if girls are interested in using computers, pre-existing habits of sex segregation can inhibit their desires. If the boys go to the computer center, then the girls may likely decline to enter there.

By male self selection and female default, the computer center becomes defined as "male turf"—as socially inappropriate to girls as the boys' locker room. One observer has pointed out that both girls and boys make equal use of computers when there is only one computer in the classroom, but girls' usage drops off when the computers are centralized in a computer center.

Social Context of Computing

Most of the messages that surround computer use are male oriented or stress mathematics. Advertisements on television and in computer magazines predominantly depict men and boys as the users of computers; girls rarely appear or appear simply as observers. One recent study of illustrations in advertisements and articles in three major computer magazines found that 69 percent of the illustrations depicted only men or boys as computer users and only 13 percent featured only girls or women. Moreover, the titles of computer games and software are usually directed to a male audience. When junior high school students were asked to evaluate the titles of 75 randomly selected pieces of software for gender orientation, the students rated nearly 40 percent of the titles as written primarily for males and only 5 percent of interest to females.

Another aspect of the context of computing that has been frequently commented on by researchers but rarely—if ever—studied is the association between computing and mathematics. All too often, computing courses are offered by teachers of mathematics in conjunction with mathematics courses. These same teachers may have been instrumental in introducing computers into the schools. Although developing programming algorithms and solving mathematical problems may both require logical thinking, it is a mistake to assume that mathematicians are the only people who find computers useful. Writers, editors, choreographers, social scientists, artists—to name but a few—all find computers valuable in their work.

Direct and Indirect Costs

There is evidence that the lower rate of female participation in computer activities may also be the consequence of the direct and indirect costs of computing. Direct costs, which include the purchase and maintenance of a computer and appropriate software, range from $200 for the simplest machine to over $4000 for a computer with substantial memory, disk drives and printer. It appears that parents are more willing to invest in a home computer and computer training for their sons than for their daughters. For example, a survey of California students found that more boys than girls reported having video games at home. Even in one high income district, twice as many boys (15%) as girls (7.5%) surveyed in 1982 reported having computers at home. A survey of computer camps found that male enrollments outnumbered female enrollments three to one, and that the more expensive the camp, the lower the female enrollments.

The indirect costs of computing include opportunity costs—that is, what the participant would have to give up in order to engage in computer-related activities. A student must invest an enormous amount of time to become fully proficient at programming. According to time/use studies of high school students, girls spend more of their free time interacting with friends than do boys. Thus, a girl who chooses to isolate herself in the computer center will incur greater opportunity costs than will a boy, since her behavior may jeopardize both her friendships and her social standing.

Quality and Relevance of the Computer Curriculum

Because of the limited scope of computer courses, girls often report that the courses are neither stimulating, interesting nor relevant. Studies show that girls find little immediate practical use for simple programming skills and would rather learn applications programs for word processing, database use or graphics. When asked about their interest in further computer-related courses, the responses of 400 students in a required computer course showed an interesting sex difference: the majority (52%) of the girls indicated an interest in word processing or business/research applications, whereas the majority of the boys (59.4%) indicated an interest in additional training in programming languages. The same study also showed other interesting sex differences: fewer girls than boys liked working with computers, learning to program, or writing programs to solve problems.

What Can Teachers Do?

Teachers of computing courses have a responsibility to be aware of and to counterbalance many of the factors that affect lower rates of female participation in computer-related activities. Initially, they can encourage female participation by making four simple changes in their classrooms and curricula.

1. **The first action teachers can take is to change the image of the computer center as "male turf."** Strategies teachers have used to encourage girls' use of the center include reserving it for girls on certain days of the week and prohibiting the playing of games. **Computer centers should lose their image as male game preserves.**

2. **Teachers should review the software in their computer centers and eliminate materials that are primarily oriented toward one sex.** Instructional packages that rely on "attacks," "hits," and other aggressive actions for success should be questioned for their educational value; perhaps software with a less aggressive message could be obtained. Also, because certain games are particularly offensive to women, a blanket prohibition against game playing (as opposed to the more difficult and problematic method of censoring specific games) can promote a more equitable and gender-fair atmosphere in the computer center. In addition, elimination of games will free up computers for their primary educational uses.

3. **While teachers can do little to reduce the direct costs of computing at home, they can provide access to computers for those students who do not have home computers, and they can demonstrate the value of computers to students engaged in other academic or extracurricular activities.** Teachers should present the computer as a valuable cost-saving and time-saving resource for many purposes, not just a medium for game-playing or programming for the sake of programming.

This approach, however, may require increasing the resources available in the center. For example, sufficient printers and word processing software should be available so that students can prepare reports and papers.

Teachers should encourage the use of the computer center by the school newspaper and literary magazine for word processing, editing and camera-ready production. Teachers should also reach out to school organizations, clubs, societies and teams and assist them in using the computers for maintaining files for such purposes as rosters, meeting schedules, fund-raising records, minutes, inter- and intramural team and individual records, budgets, voting records or attendance. Not only would this be an optimal utilization of the computer center, it would demonstrate to the students the wide range of computer applications. It would also unquestionably involve many more girls in computer-related activities by establishing a direct link between their other social and extracurricular activities and computer technology.

4. **Finally, teachers need to revise their computer course curricula. Introductory courses should stress applications programs, not computer programming.** Introductory courses could explore these types of applications programs: **word processing programs** such as Bank Street Writer for younger children and Wordstar or Applewriter II for older students; **personal filing systems**, such as PFS, for creating bibliographies, records of cookie sales and ticket sales; **integrated systems**, such as The Incredible Jack, for preparing reports; and **spread sheets**, such as Visi-Calc, for database management. If introductory computing courses present the usefulness of computers rather than the mechanics of computing, it is reasonable to asume that issues of equity and computers would not be as great in the future as they are in the present. END■

[Dr. Marlaine E. Lockheed, Senior Research Scientist, Division of Education Policy Research and Services, Educational Testing Service, Princeton, NJ 08541; Mr. Steven B. Frakt, 41 Leabrook Lane, Princeton, NJ 08540.]

References

Becker, H.J. (1982) National Survey of School Uses of Microcomputers, Interim Reports 1-6. Johns Hopkins University, Baltimore, MD.

Fetler, M. (1983) Comptuer Literacy of California Sixth and Twelfth Grade Students. California Department of Education, Division of Planning Evaluation and Research, Sacramento, CA.

Lockheed, M.E. (In press) Sex equity in classroom organization and climate. In S. Klein (Ed.) *A handbook for achieving sex equity in education.* Baltimore: Johns Hopkins University Press.

Lockheed, M.E., Nielsen, A., & Stone, M. (1983) Sex differences in microcomputer literacy. In D. Bonnette (Ed.) *Proceedings of NECC/5 National Educational Computing Conference 1983*, IEEE Computer Society Press.

Lockheed, M.E., Nielsen, A., & Stone, M. (1983) *Some determinants of microcomputer literacy.* Paper presented at the annual meeting of the American Educational Research Association, Toronto.

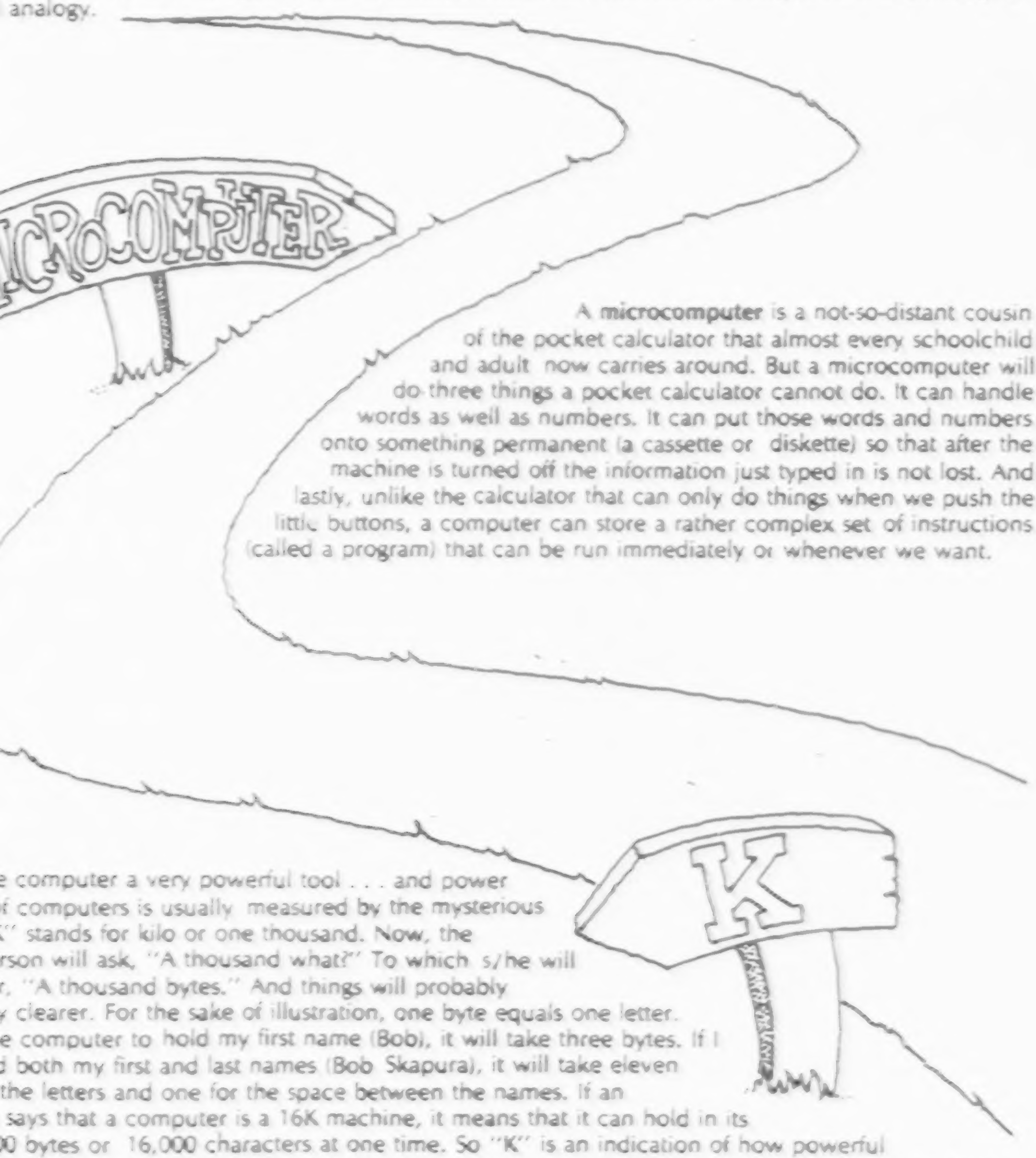
Miura, I. & Hess, R.D. (1983) *Sex differences in computer access, interest and usage.* Paper presented at the annual meetings of the American Psychological Association, Anaheim, CA.

Ware, M.C. & Stuck, M.F. (1983) Sex role messages vis-a-vis microcomputer use: A look at the pictures. Manuscript submitted for publication.

A Nod to the Novice #1

by Bob Skapura

nesbury sent two of his cartoon characters into a computer store for the first time, he gave one of
e book as though the character were entering a foreign country, anticipating that he would have to
n language. The humor lay in how close he was to the truth. This column is meant for the beginner.
vill be more of a travelogue than a dictionary. A dictionary defines; a travelogue describes through il-
analogy.



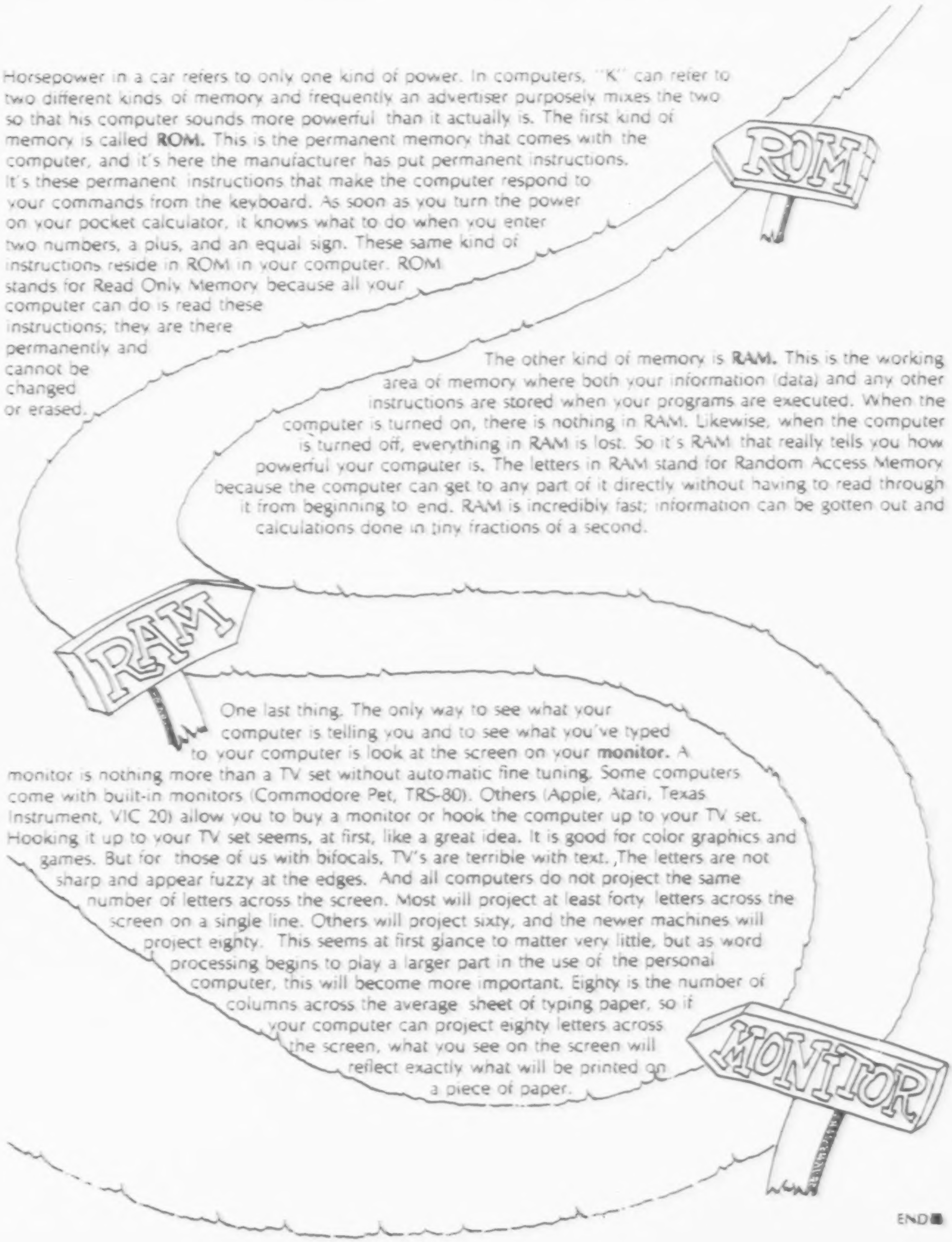
A **microcomputer** is a not-so-distant cousin of the pocket calculator that almost every schoolchild and adult now carries around. But a microcomputer will do three things a pocket calculator cannot do. It can handle words as well as numbers. It can put those words and numbers onto something permanent (a cassette or diskette) so that after the machine is turned off the information just typed in is not lost. And lastly, unlike the calculator that can only do things when we push the little buttons, a computer can store a rather complex set of instructions (called a program) that can be run immediately or whenever we want.

e computer a very powerful tool . . . and power
f computers is usually measured by the mysterious
" stands for kilo or one thousand. Now, the
rson will ask, "A thousand what?" To which s/he will
r, "A thousand bytes." And things will probably
y clearer. For the sake of illustration, one byte equals one letter.
e computer to hold my first name (Bob), it will take three bytes. If I
d both my first and last names (Bob Skapura), it will take eleven
the letters and one for the space between the names. If an
says that a computer is a 16K machine, it means that it can hold in its
0 bytes or 16,000 characters at one time. So "K" is an indication of how powerful
, something like horsepower is to a car. A 48K machine is more powerful than a 16K machine.
rs ago 16K was standard; now 64K is becoming average for microcomputers and additional
K units) can be added for as little as $50.

Horsepower in a car refers to only one kind of power. In computers, "K" can refer to two different kinds of memory and frequently an advertiser purposely mixes the two so that his computer sounds more powerful than it actually is. The first kind of memory is called **ROM.** This is the permanent memory that comes with the computer, and it's here the manufacturer has put permanent instructions. It's these permanent instructions that make the computer respond to your commands from the keyboard. As soon as you turn the power on your pocket calculator, it knows what to do when you enter two numbers, a plus, and an equal sign. These same kind of instructions reside in ROM in your computer. ROM stands for Read Only Memory because all your computer can do is read these instructions; they are there permanently and cannot be changed or erased.

The other kind of memory is **RAM.** This is the working area of memory where both your information (data) and any other instructions are stored when your programs are executed. When the computer is turned on, there is nothing in RAM. Likewise, when the computer is turned off, everything in RAM is lost. So it's RAM that really tells you how powerful your computer is. The letters in RAM stand for Random Access Memory because the computer can get to any part of it directly without having to read through it from beginning to end. RAM is incredibly fast; information can be gotten out and calculations done in tiny fractions of a second.

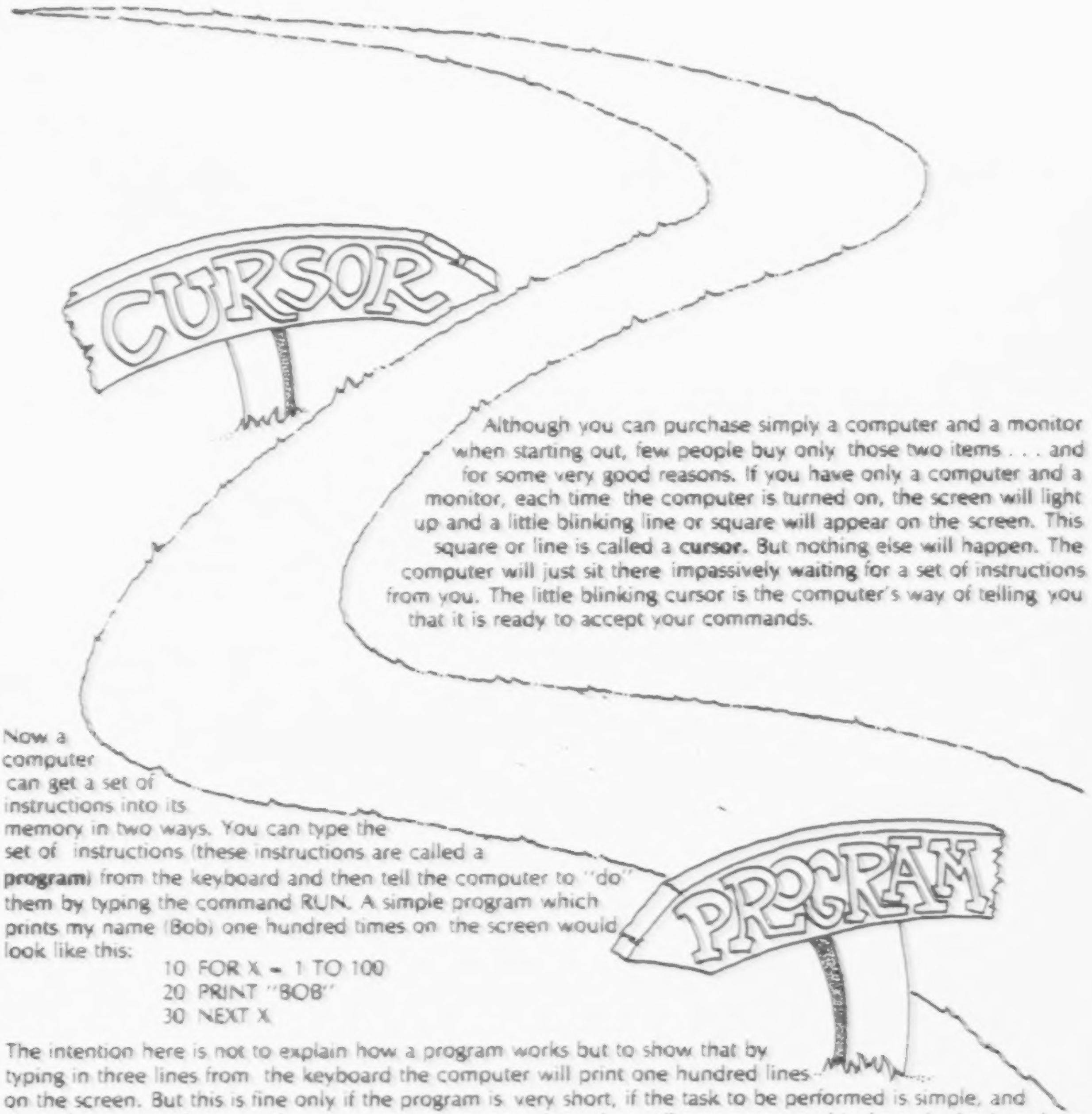
One last thing. The only way to see what your computer is telling you and to see what you've typed to your computer is look at the screen on your **monitor.** A monitor is nothing more than a TV set without automatic fine tuning. Some computers come with built-in monitors (Commodore Pet, TRS-80). Others (Apple, Atari, Texas Instrument, VIC 20) allow you to buy a monitor or hook the computer up to your TV set. Hooking it up to your TV set seems, at first, like a great idea. It is good for color graphics and games. But for those of us with bifocals, TV's are terrible with text. The letters are not sharp and appear fuzzy at the edges. And all computers do not project the same number of letters across the screen. Most will project at least forty letters across the screen on a single line. Others will project sixty, and the newer machines will project eighty. This seems at first glance to matter very little, but as word processing begins to play a larger part in the use of the personal computer, this will become more important. Eighty is the number of columns across the average sheet of typing paper, so if your computer can project eighty letters across the screen, what you see on the screen will reflect exactly what will be printed on a piece of paper.



END■

A Nod to the Novice #2

by Bob Skapura



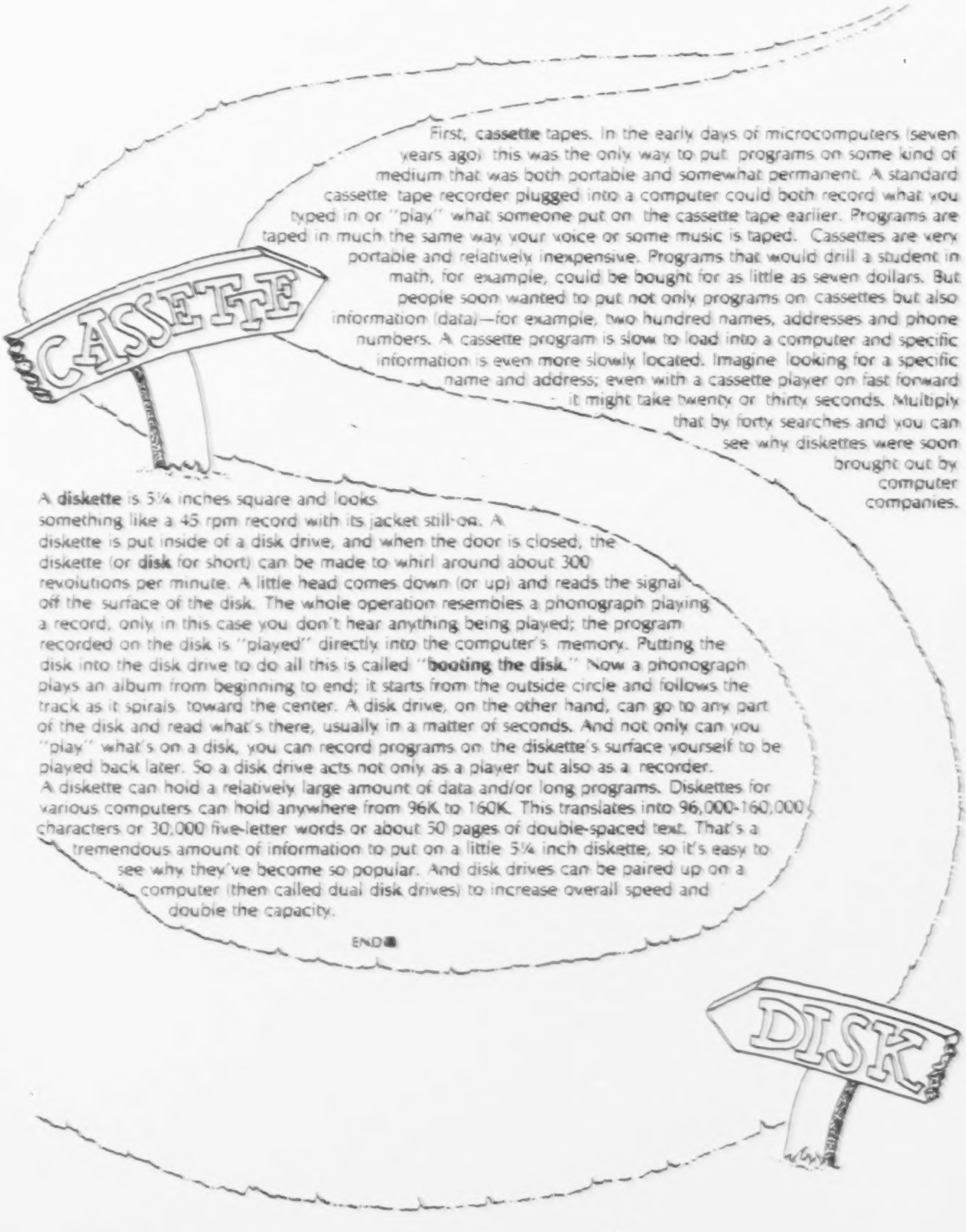
Although you can purchase simply a computer and a monitor when starting out, few people buy only those two items . . . and for some very good reasons. If you have only a computer and a monitor, each time the computer is turned on, the screen will light up and a little blinking line or square will appear on the screen. This square or line is called a **cursor.** But nothing else will happen. The computer will just sit there impassively waiting for a set of instructions from you. The little blinking cursor is the computer's way of telling you that it is ready to accept your commands.

Now a computer can get a set of instructions into its memory in two ways. You can type the set of instructions (these instructions are called a **program**) from the keyboard and then tell the computer to "do" them by typing the command RUN. A simple program which prints my name (Bob) one hundred times on the screen would look like this:

```
10 FOR X = 1 TO 100
20 PRINT "BOB"
30 NEXT X
```

The intention here is not to explain how a program works but to show that by typing in three lines from the keyboard the computer will print one hundred lines on the screen. But this is fine only if the program is very short, if the task to be performed is simple, and if you have no wish to run the same program at some later date. All programs typed in from the keyboard are placed in RAM, the part of the computer's memory that is erased when the machine is shut off. The problem is that most tasks are not simple, the programs are not short, and few people are willing to spend hours typing in hundreds of program lines which would take only a few moments to run and then will be erased when the computer is shut off.

The second way of getting the set of instructions into a computer's memory overcomes most of these difficulties. Programs can be "read into" a computer from a cassette or diskette. These programs are originally typed in from some keyboard and then saved onto either cassette or diskette before the computer is shut off; these programs can be put on cassette or diskette by you or by some professional programmer.



First, **cassette** tapes. In the early days of microcomputers (seven years ago) this was the only way to put programs on some kind of medium that was both portable and somewhat permanent. A standard cassette tape recorder plugged into a computer could both record what you typed in or "play" what someone put on the cassette tape earlier. Programs are taped in much the same way your voice or some music is taped. Cassettes are very portable and relatively inexpensive. Programs that would drill a student in math, for example, could be bought for as little as seven dollars. But people soon wanted to put not only programs on cassettes but also information (data)—for example, two hundred names, addresses and phone numbers. A cassette program is slow to load into a computer and specific information is even more slowly located. Imagine looking for a specific name and address; even with a cassette player on fast forward it might take twenty or thirty seconds. Multiply that by forty searches and you can see why diskettes were soon brought out by computer companies.

A **diskette** is 5¼ inches square and looks something like a 45 rpm record with its jacket still-on. A diskette is put inside of a disk drive, and when the door is closed, the diskette (or **disk** for short) can be made to whirl around about 300 revolutions per minute. A little head comes down (or up) and reads the signal off the surface of the disk. The whole operation resembles a phonograph playing a record, only in this case you don't hear anything being played; the program recorded on the disk is "played" directly into the computer's memory. Putting the disk into the disk drive to do all this is called "**booting the disk.**" Now a phonograph plays an album from beginning to end; it starts from the outside circle and follows the track as it spirals toward the center. A disk drive, on the other hand, can go to any part of the disk and read what's there, usually in a matter of seconds. And not only can you "play" what's on a disk, you can record programs on the diskette's surface yourself to be played back later. So a disk drive acts not only as a player but also as a recorder.

A diskette can hold a relatively large amount of data and/or long programs. Diskettes for various computers can hold anywhere from 96K to 160K. This translates into 96,000-160,000 characters or 30,000 five-letter words or about 50 pages of double-spaced text. That's a tremendous amount of information to put on a little 5¼ inch diskette, so it's easy to see why they've become so popular. And disk drives can be paired up on a computer (then called dual disk drives) to increase overall speed and double the capacity.

END■

A Nod to the Novice #3

by
Robert Skapura

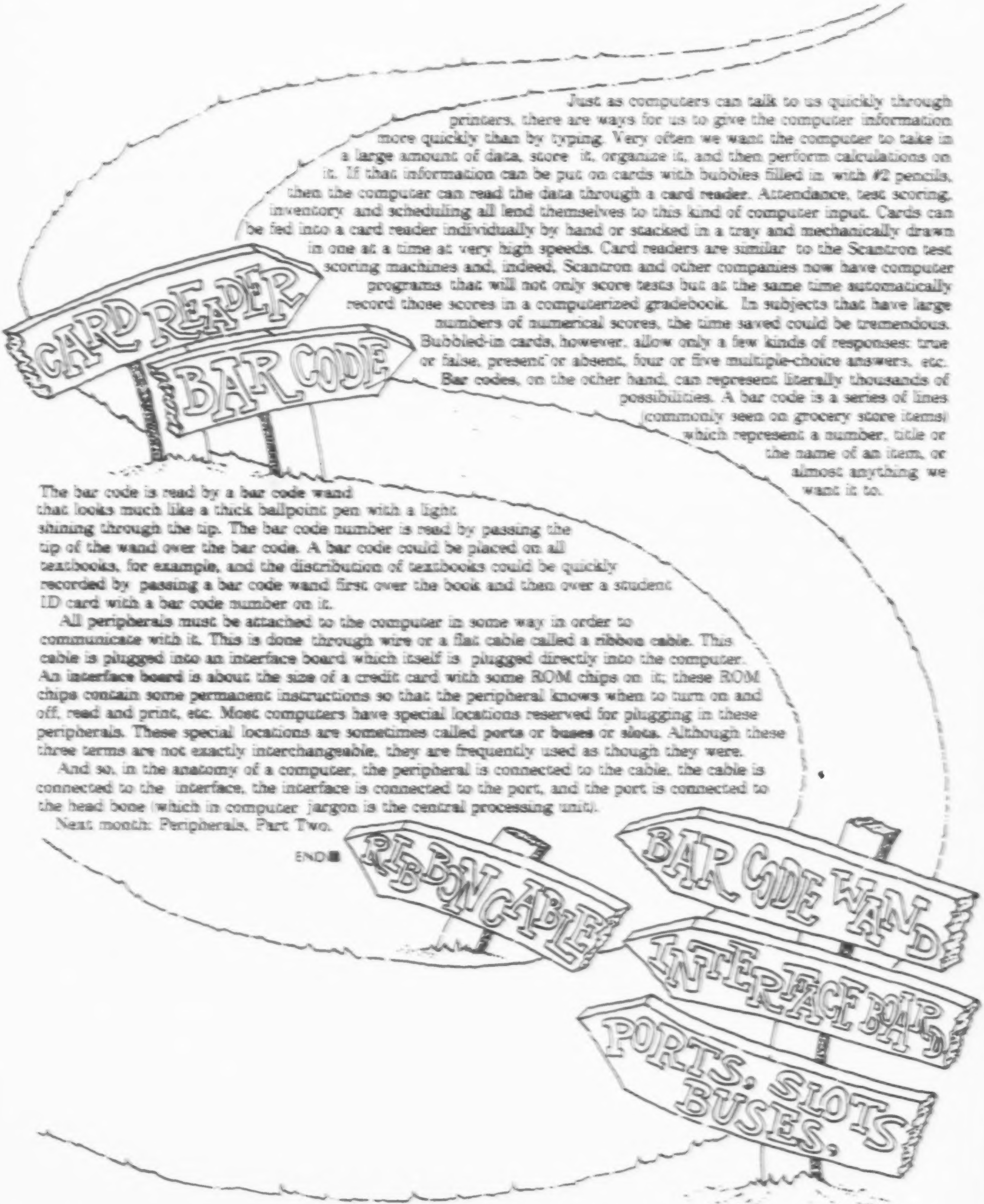
PERIPHERALS
PRINTER

To teach programming to students, all that is needed is a computer, monitor, and perhaps a disk drive or cassette to either store student written programs or to run commercially developed ones. But the real power of the microcomputer lies not in these three pieces of hardware but in all the other things that can be attached to micros to make them serve many special and very useful functions. These other things that can plug into computers are called **peripherals**.

Probably the first peripheral bought by most people is a printer. Printers are divided into three basic categories: thermal, dot matrix and letter quality. **Thermal printers** produce letters on special paper. They're sometimes called silent printers because they quietly produce an image on paper by passing a heated element over a special oxide- treated paper. The printers themselves are inexpensive, but the paper costs are relatively high. Probably the most common kind of printer in schools today is the **dot matrix**. These printers got this name because letters (and other images) are produced by a series of closely configured dots.

The chief criticism of dot matrix printers is that the net product looks "dot matrix." However, they're fast, inexpensive to operate, and when used for producing lists and other information, are more than adequate. A step up in both price and appearance is the **letter quality printer**. It obviously takes its name from the look it gives the printed product. Letter quality printers produce letters in much the same way as electronic typewriters do—a hammer strikes a letter which strikes the paper through a ribbon. The result--copy that resembles that produced by a high quality typewriter. A standard typewriter can print twelve characters a second at top speed--printers under the control of a computer can produce anywhere from 25 to 200 characters a second! In many schools, letter quality printers are being used in conjunction with word processing programs to produce camera-ready copy for school newspapers, parent newsletters, announcements and schedules. The speed and professional look of these printed materials could be achieved in the past only by going outside the school to typesetters and print shops.

If no peripherals are attached to the computer, there is only one way for people to "talk" to it. The user must patiently type one word or command at a time on the keyboard, and the computer can reply only through the screen of the monitor. Peripherals, on the other hand, allow for the computer and the outside world to make contact in other ways. Other peripherals are not yet as common as printers, but they are equally promising in their usefulness to educators.



Just as computers can talk to us quickly through printers, there are ways for us to give the computer information more quickly than by typing. Very often we want the computer to take in a large amount of data, store it, organize it, and then perform calculations on it. If that information can be put on cards with bubbles filled in with #2 pencils, then the computer can read the data through a **card reader**. Attendance, test scoring, inventory and scheduling all lend themselves to this kind of computer input. Cards can be fed into a card reader individually by hand or stacked in a tray and mechanically drawn in one at a time at very high speeds. Card readers are similar to the Scantron test scoring machines and, indeed, Scantron and other companies now have computer programs that will not only score tests but at the same time automatically record those scores in a computerized gradebook. In subjects that have large numbers of numerical scores, the time saved could be tremendous. Bubbled-in cards, however, allow only a few kinds of responses: true or false, present or absent, four or five multiple-choice answers, etc. **Bar codes**, on the other hand, can represent literally thousands of possibilities. A bar code is a series of lines (commonly seen on grocery store items) which represent a number, title or the name of an item, or almost anything we want it to.

The bar code is read by a **bar code wand** that looks much like a thick ballpoint pen with a light shining through the tip. The bar code number is read by passing the tip of the wand over the bar code. A bar code could be placed on all textbooks, for example, and the distribution of textbooks could be quickly recorded by passing a bar code wand first over the book and then over a student ID card with a bar code number on it.

All peripherals must be attached to the computer in some way in order to communicate with it. This is done through wire or a flat cable called a **ribbon cable**. This cable is plugged into an interface board which itself is plugged directly into the computer. An **interface board** is about the size of a credit card with some ROM chips on it; these ROM chips contain some permanent instructions so that the peripheral knows when to turn on and off, read and print, etc. Most computers have special locations reserved for plugging in these peripherals. These special locations are sometimes called **ports** or **buses** or **slots**. Although these three terms are not exactly interchangeable, they are frequently used as though they were.

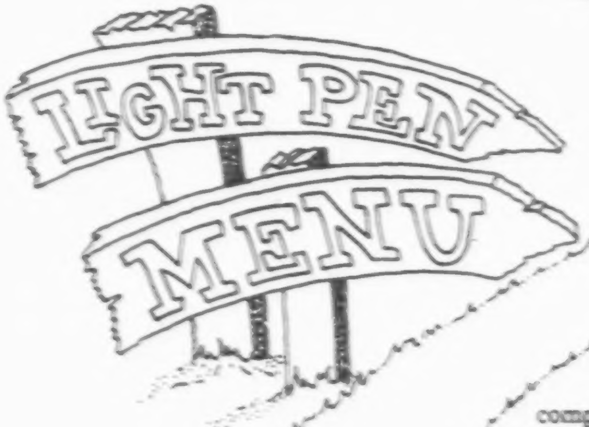
And so, in the anatomy of a computer, the peripheral is connected to the cable, the cable is connected to the interface, the interface is connected to the port, and the port is connected to the head bone (which in computer jargon is the central processing unit).

Next month: Peripherals, Part Two.

END■

A Nod to the Novice #4: Peripherals, Part 2

by
Robert Skapura

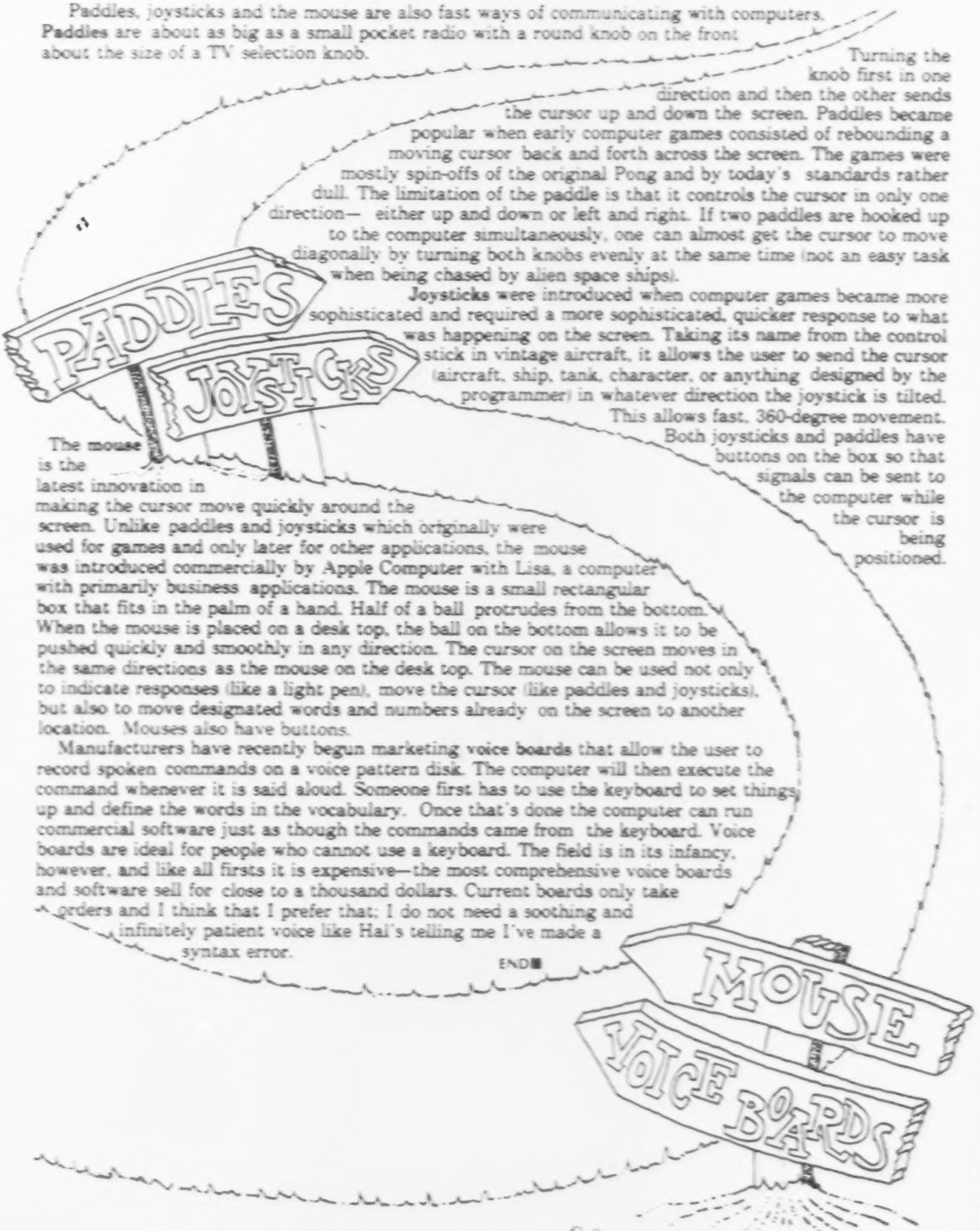


People talk to computers by typing on the computer keyboard; computers talk to people through the words and images that appear on the screen. We saw last month that printers offer a fast way for the computer to set down on paper what is in its memory or on disk. And both card readers and bar code wands give users a way to get information into the computer more quickly than using the keyboard. There are other ways for information to pass between humans and computers—not all of them so common, but all promising in their effectiveness and ease of use.

Similar in appearance to a bar code wand (but not in use) is a **light pen.** It resembles a rather thick ball point pen with a little light shining out of the bottom and a cord leading out of the top and into the computer. Plugged into a port or interface card, light pens are used most frequently to tell the computer which menu option or answer has been chosen. Most interactive educational programs present the user with a **menu,** much like a short restaurant menu. This list of choices is usually numbered or lettered so that users can indicate their choices by simply typing the appropriate number or letter.

For younger children, the handicapped, or people who aren't touch typists, this procedure can slow down their responses as they look for the letter or number on the keyboard. If the program is written to look for a sharp, intense light on the screen, a light pen can be used to touch the choice on the screen, sending the computer off to perform the next task or present the next problem.

Many drill and practice programs are put into a game format and students accumulate points, fight off monsters or evade space ships by entering correct answers to questions. I've seen young children quickly figure out a problem in their heads only to get blown up, caught or gobbled by creatures as they struggle to find the right key. This argues well that knowing the keyboard configuration is a "basic skill", but for some students it places one more obstacle in the way of learning the task at hand. There are other programs that remember the path of the light pen across the screen, so that the light pen can be used to sketch an image that can be stored in the computer's memory or printed by a printer. When you consider that most light pens cost under forty dollars, there seems to be a bright future for this peripheral.



Paddles, joysticks and the mouse are also fast ways of communicating with computers. **Paddles** are about as big as a small pocket radio with a round knob on the front about the size of a TV selection knob. Turning the knob first in one direction and then the other sends the cursor up and down the screen. Paddles became popular when early computer games consisted of rebounding a moving cursor back and forth across the screen. The games were mostly spin-offs of the original Pong and by today's standards rather dull. The limitation of the paddle is that it controls the cursor in only one direction— either up and down or left and right. If two paddles are hooked up to the computer simultaneously, one can almost get the cursor to move diagonally by turning both knobs evenly at the same time (not an easy task when being chased by alien space ships).

Joysticks were introduced when computer games became more sophisticated and required a more sophisticated, quicker response to what was happening on the screen. Taking its name from the control stick in vintage aircraft, it allows the user to send the cursor (aircraft, ship, tank, character, or anything designed by the programmer) in whatever direction the joystick is tilted. This allows fast, 360-degree movement. Both joysticks and paddles have buttons on the box so that signals can be sent to the computer while the cursor is being positioned.

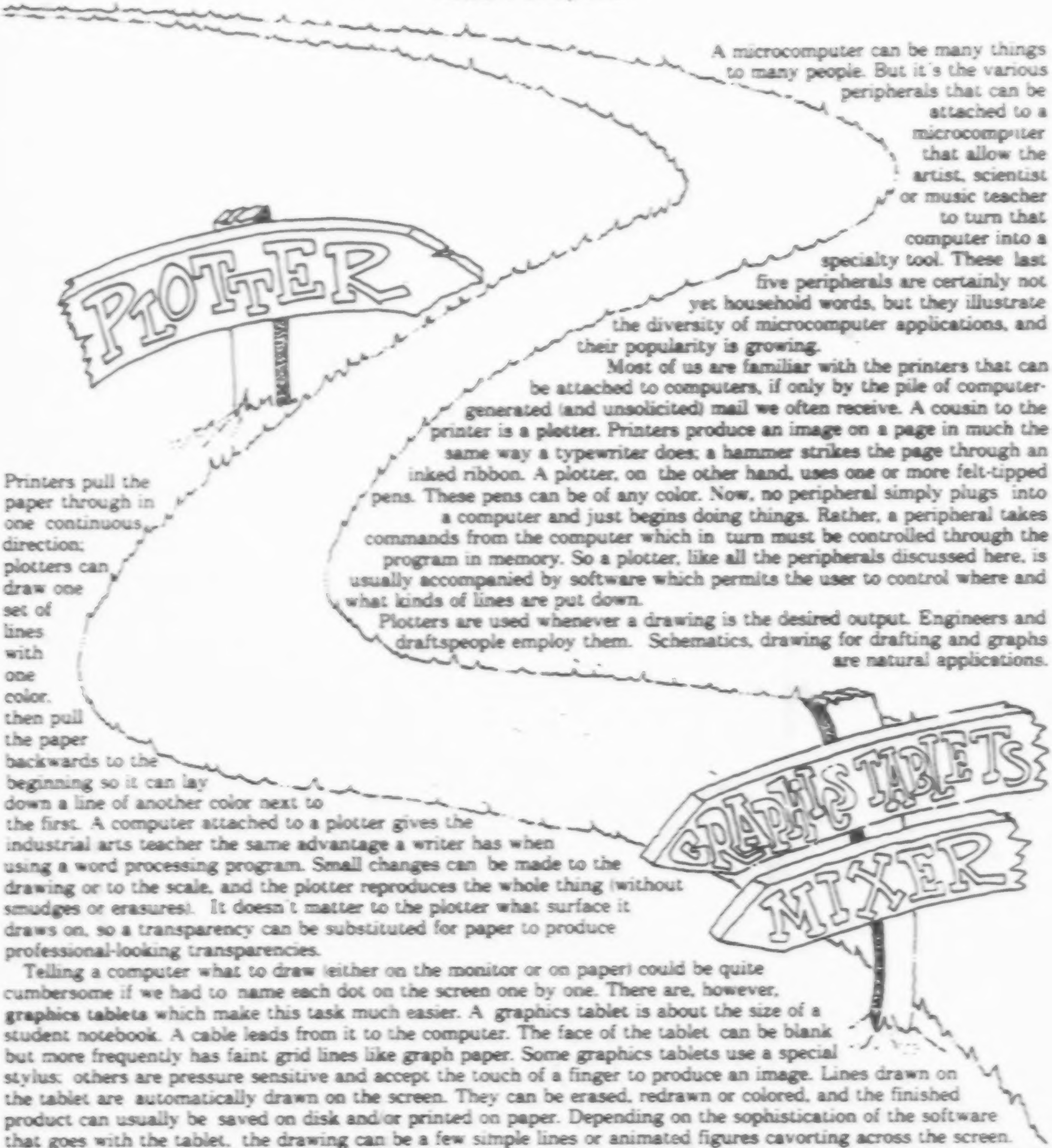
The **mouse** is the latest innovation in making the cursor move quickly around the screen. Unlike paddles and joysticks which originally were used for games and only later for other applications, the mouse was introduced commercially by Apple Computer with Lisa, a computer with primarily business applications. The mouse is a small rectangular box that fits in the palm of a hand. Half of a ball protrudes from the bottom. When the mouse is placed on a desk top, the ball on the bottom allows it to be pushed quickly and smoothly in any direction. The cursor on the screen moves in the same directions as the mouse on the desk top. The mouse can be used not only to indicate responses (like a light pen), move the cursor (like paddles and joysticks), but also to move designated words and numbers already on the screen to another location. Mouses also have buttons.

Manufacturers have recently begun marketing **voice boards** that allow the user to record spoken commands on a voice pattern disk. The computer will then execute the command whenever it is said aloud. Someone first has to use the keyboard to set things up and define the words in the vocabulary. Once that's done the computer can run commercial software just as though the commands came from the keyboard. Voice boards are ideal for people who cannot use a keyboard. The field is in its infancy, however, and like all firsts it is expensive—the most comprehensive voice boards and software sell for close to a thousand dollars. Current boards only take orders and I think that I prefer that; I do not need a soothing and infinitely patient voice like Hal's telling me I've made a syntax error.

END■

A Nod to the Novice #5: Peripherals, Part 3

by
Robert Skapura



A microcomputer can be many things to many people. But it's the various peripherals that can be attached to a microcomputer that allow the artist, scientist or music teacher to turn that computer into a specialty tool. These last five peripherals are certainly not yet household words, but they illustrate the diversity of microcomputer applications, and their popularity is growing.

Most of us are familiar with the printers that can be attached to computers, if only by the pile of computer-generated (and unsolicited) mail we often receive. A cousin to the printer is a **plotter**. Printers produce an image on a page in much the same way a typewriter does: a hammer strikes the page through an inked ribbon. A plotter, on the other hand, uses one or more felt-tipped pens. These pens can be of any color. Now, no peripheral simply plugs into a computer and just begins doing things. Rather, a peripheral takes commands from the computer which in turn must be controlled through the program in memory. So a plotter, like all the peripherals discussed here, is usually accompanied by software which permits the user to control where and what kinds of lines are put down.

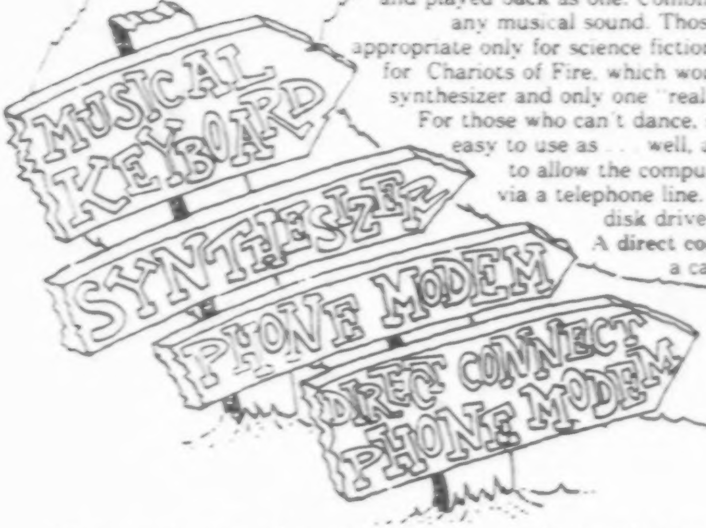
Plotters are used whenever a drawing is the desired output. Engineers and draftspeople employ them. Schematics, drawing for drafting and graphs are natural applications. Printers pull the paper through in one continuous direction; plotters can draw one set of lines with one color, then pull the paper backwards to the beginning so it can lay down a line of another color next to the first. A computer attached to a plotter gives the industrial arts teacher the same advantage a writer has when using a word processing program. Small changes can be made to the drawing or to the scale, and the plotter reproduces the whole thing (without smudges or erasures). It doesn't matter to the plotter what surface it draws on, so a transparency can be substituted for paper to produce professional-looking transparencies.

Telling a computer what to draw (either on the monitor or on paper) could be quite cumbersome if we had to name each dot on the screen one by one. There are, however, **graphics tablets** which make this task much easier. A graphics tablet is about the size of a student notebook. A cable leads from it to the computer. The face of the tablet can be blank but more frequently has faint grid lines like graph paper. Some graphics tablets use a special stylus; others are pressure sensitive and accept the touch of a finger to produce an image. Lines drawn on the tablet are automatically drawn on the screen. They can be erased, redrawn or colored, and the finished product can usually be saved on disk and/or printed on paper. Depending on the sophistication of the software that goes with the tablet, the drawing can be a few simple lines or animated figures cavorting across the screen.

For the artist with a musical bent, there are even more powerful peripherals, although these are harder to use and more expensive. They do open worlds not available just a few years ago. Starting with a special interface board or two, the computer can be connected to a **mixer** or to stereo speakers. There are software packages that display a musical staff on the screen and will play whatever notes are put there. Commands can come from the computer keyboard, a set of paddles or a light pen. You can specify the clef, time and the key of the composition.

You can also make choices about duration of individual chords, notes and rests, measure placement and accidentals. A **musical keyboard** or clavier, similar to a piano or small organ, can be attached directly to the computer. A composition can be created and saved directly on disk, and the sheet music then can be printed on the printer. (Some software requires special printer interface cards.) Different parts and "voices" can be composed separately and played back as one. Combined with a **synthesizer**, a computer can produce almost any musical sound. Those who feel that "synthesized" music sounds eerie or is appropriate only for science fiction or horror movies should keep in mind that the music for Chariots of Fire, which won an Academy Awared, was produced by a computer, a synthesizer and only one "real" piano.

For those who can't dance, sing or carry a tune, there are still other peripherals as easy to use as . . . well, a telephone. In fact, the purpose of a **phone modem** is to allow the computer to communicate to the outside world just as we do via a telephone line. Phone modems are generally smaller than the average disk drive and come in two kinds: direct connect and acoustical. A **direct connect phone modem** plugs into an interface board with a cable that looks like the one that snaps into the back of the typical phone. Software can be purchased that will activate the modem and allow it to either transmit to or receive from other computers.

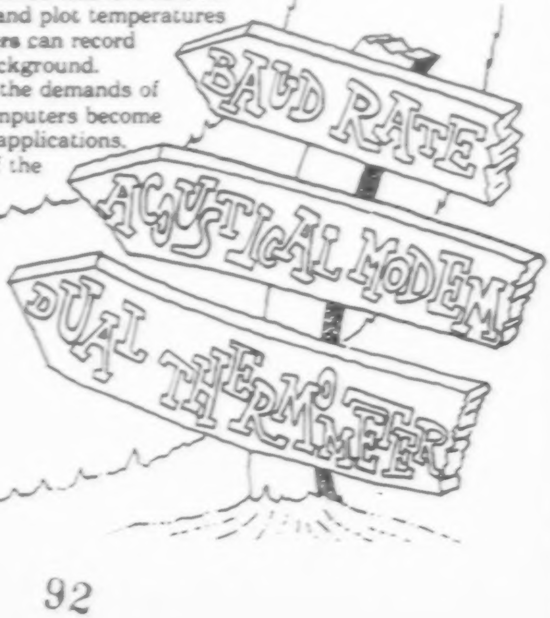


It can send messages typed directly from the keyboard, somewhat like an old teletype machine, or it can transmit data from disk or memory to another computer in just a few seconds. The rate that information is sent through telephone lines is called the **baud rate**. The minimum baud rate is 300 baud (about forty characters a second); 1200 baud is not uncommon. An **acoustical modem**, or acoustic coupler, is also called a phone cradle because the user's receiver is taken off the hook and placed on a cradle that cups the mouth and earpiece and transmits the "touch tone" sound to activate the phone. Attaching a phone modem to a microcomputer allows the user to communicate to other computer users almost anywhere to exchange information and to tap larger information data bases like The Source and Dialog.

From all these examples, it becomes obvious that almost anything electrical can be attached to a computer to send, receive, generate, record and modify signals. Science teachers can, for example, attach a **dual thermometer** and an internal clock to record and plot temperatures during an experiment over long periods of time. Likewise, **photometers** can record and plot the speed and acceleration of lighted objects past a dark background. Presently there are not even enough computers in schools to satisfy the demands of the students, and those requirements should be met first. But as computers become more numerous and teachers in all fields experiment with their applications, they will find that the only limitations will be the length of the electric cord and the location of the phone line.

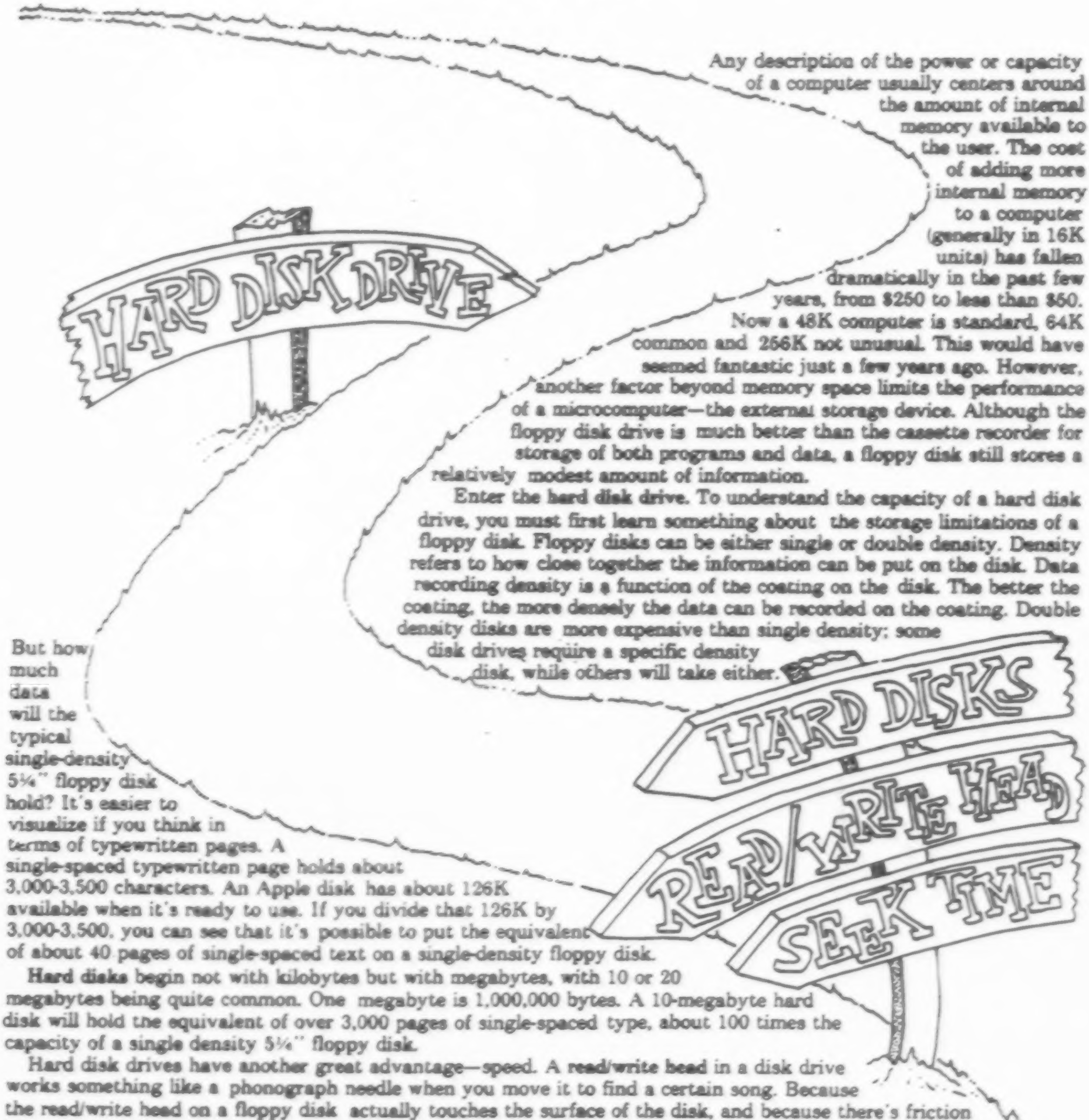
END■

PHOTOMETERS



A Nod to the Novice #6: Peripherals, Part 4

by
Robert Skapura



Any description of the power or capacity of a computer usually centers around the amount of internal memory available to the user. The cost of adding more internal memory to a computer (generally in 16K units) has fallen dramatically in the past few years, from $250 to less than $50. Now a 48K computer is standard, 64K common and 256K not unusual. This would have seemed fantastic just a few years ago. However, another factor beyond memory space limits the performance of a microcomputer—the external storage device. Although the floppy disk drive is much better than the cassette recorder for storage of both programs and data, a floppy disk still stores a relatively modest amount of information.

Enter the **hard disk drive.** To understand the capacity of a hard disk drive, you must first learn something about the storage limitations of a floppy disk. Floppy disks can be either single or double density. Density refers to how close together the information can be put on the disk. Data recording density is a function of the coating on the disk. The better the coating, the more densely the data can be recorded on the coating. Double density disks are more expensive than single density; some disk drives require a specific density disk, while others will take either.

But how much data will the typical single-density 5¼" floppy disk hold? It's easier to visualize if you think in terms of typewritten pages. A single-spaced typewritten page holds about 3,000-3,500 characters. An Apple disk has about 126K available when it's ready to use. If you divide that 126K by 3,000-3,500, you can see that it's possible to put the equivalent of about 40 pages of single-spaced text on a single-density floppy disk.

Hard disks begin not with kilobytes but with megabytes, with 10 or 20 megabytes being quite common. One megabyte is 1,000,000 bytes. A 10-megabyte hard disk will hold the equivalent of over 3,000 pages of single-spaced type, about 100 times the capacity of a single density 5¼" floppy disk.

Hard disk drives have another great advantage—speed. A **read/write head** in a disk drive works something like a phonograph needle when you move it to find a certain song. Because the read/write head on a floppy disk actually touches the surface of the disk, and because there's friction between the disk and the square envelope it's in, there's a practical limit to the speed it can rotate—usually about 300 rpm. Hard disks, however, spin at 3,600 rpm. This speed dramatically reduces the time it takes for the drive to find a specific point on the disk. This is called **seek time.** While a floppy disk drive's seek time is between 1/10 and 1 second, most hard disk drives have a seek time of 50 thousandths of a second. This means that a typical hard disk drive for a microcomputer can find and load a 64K file in less than a second—barely longer than it takes the computer to clear the screen and display a new page of text.

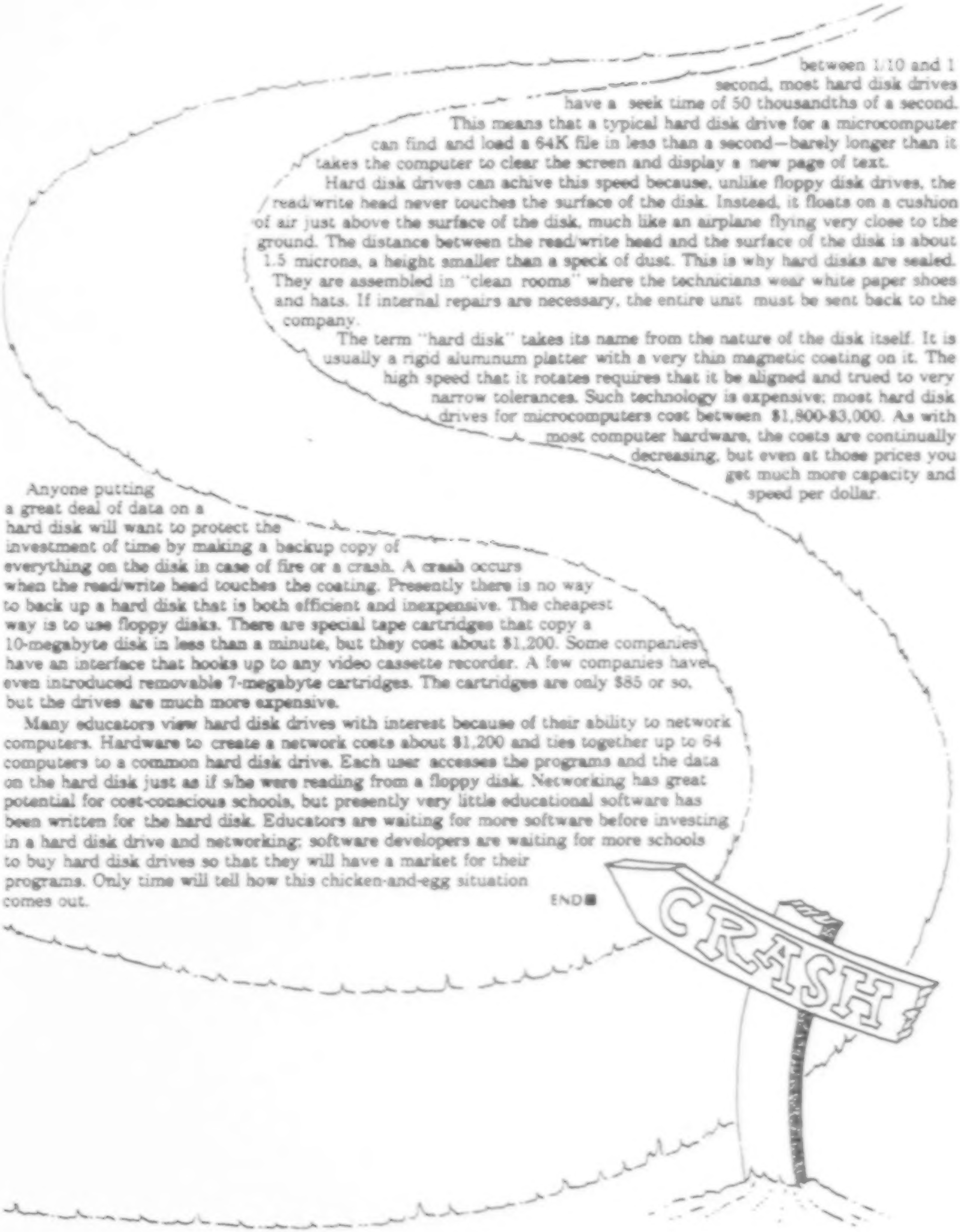
Hard disk drives can achive this speed because, unlike floppy disk drives, the read/write head never touches the surface of the disk. Instead, it floats on a cushion of air just above the surface of the disk, much like an airplane flying very close to the ground. The distance between the read/write head and the surface of the disk is about 1.5 microns, a height smaller than a speck of dust. This is why hard disks are sealed. They are assembled in "clean rooms" where the technicians wear white paper shoes and hats. If internal repairs are necessary, the entire unit must be sent back to the company.

The term "hard disk" takes its name from the nature of the disk itself. It is usually a rigid aluminum platter with a very thin magnetic coating on it. The high speed that it rotates requires that it be aligned and trued to very narrow tolerances. Such technology is expensive; most hard disk drives for microcomputers cost between $1,800-$3,000. As with most computer hardware, the costs are continually decreasing, but even at those prices you get much more capacity and speed per dollar.

Anyone putting a great deal of data on a hard disk will want to protect the investment of time by making a backup copy of everything on the disk in case of fire or a crash. A crash occurs when the read/write head touches the coating. Presently there is no way to back up a hard disk that is both efficient and inexpensive. The cheapest way is to use floppy disks. There are special tape cartridges that copy a 10-megabyte disk in less than a minute, but they cost about $1,200. Some companies have an interface that hooks up to any video cassette recorder. A few companies have even introduced removable 7-megabyte cartridges. The cartridges are only $85 or so, but the drives are much more expensive.

Many educators view hard disk drives with interest because of their ability to network computers. Hardware to create a network costs about $1,200 and ties together up to 64 computers to a common hard disk drive. Each user accesses the programs and the data on the hard disk just as if s/he were reading from a floppy disk. Networking has great potential for cost-conscious schools, but presently very little educational software has been written for the hard disk. Educators are waiting for more software before investing in a hard disk drive and networking; software developers are waiting for more schools to buy hard disk drives so that they will have a market for their programs. Only time will tell how this chicken-and-egg situation comes out.

END■



The International Council for Computers in Education (ICCE)

The International Council for Computers in Education (ICCE) is a non-profit educational organization with 15,000 Individual Members and 43 Organization Members.

ICCE publishes *The Computing Teacher* as well as many booklets for persons interested in the instructional use of computers at the pre-college level. All ICCE publications emphasize teaching about computers, teaching using computers, teacher education, and the impact of computers on curriculum.

TCT includes regular features and columns such as Software and Book Reviews, Computers in the Teaching of English, The Logo Center, Computers in the Media Center, Computers in the Arts and Humanities, and Computers in Science Education. TCT emphasizes practical, in-depth articles that are useful for K-12 teachers. Back issues of *The Computing Teacher* are available. Several issues of TCT have been special theme issues:

Vol. 10 #3—Logo
Vol. 10 #6—Special Needs
Vol. 11 #5—Logo
Vol. 11 #7—Public Domain Software
Vol. 11 #8—Equity

ICCE has a substantial and growing professional outreach program. Five major components include:

1. **Organization Members.** Organization Members publish newsletters and/or journals, hold conferences and directly interact with their own members.
2. **Professional Staff.** ICCE has a professional staff who write, edit, participate in conferences, process orders, consult by phone or mail, etc.
3. **Ad Hoc Committees.** Such a committee recently created the "ICCE Policy Statement on Network and Multiple Machine Software" (see TCT, Sept. 1983). A committee is currently working on certification requirements for computer-using educators.
4. **Technical Liaison Committees.** These are small informal committees designed to interface between ICCE and the professional societies of other disciplines. Currently there are TLCs in math, science, English as a second language and school psychology.
5. **National Educational Computing Conference.** ICCE is one of the sponsoring societies for NECC. This year ICCE is responsible for three panels: Certification Recommendations for Computer-using Educators; Impact of ICCE Policy Statement on Network and Multiple Machine Software; Logo in the Schools: Are There Problems? (double session co-sponsored with the National Logo Exchange).
6. **Special Interest Groups (SIGs).** ICCE has taken the initial steps in organizing special interest groups to facilitate cooperation among computer educators with a specialized interest or set of needs. SIGS, along with a planned *SIG Bulletin*, will attempt to serve as a forum for educators to clarify roles, receive and exchange pertinent information, and establish and build professional identity. Four groups have been identified within the computers-in-education field as having new and challenging responsibilities:

   • Computer Coordinators. This SIG is for people who have building-level or district-level responsibility for computers in education.

   • Teachers of Educators. This SIG is for people who develop and teach courses and workshops for inservice or preservice educators.

   • Administrators. This SIG is for school administrators who need considerable computer knowledge to make effective decisions about school management and the integration of computers into the curriculum.

   • Special Educators. This SIG is for educators and parents involved in the education of students with special needs.

Initially, one *SIG Bulletin* is being created to serve all the newly emerging SIGs. Eventually, individual SIG publications will replace this combined *SIG Bulletin*.

The *SIG Bulletin*, a quarterly publication, will carry information about the SIGs that are now forming. It will contain topics of interest to each SIG with a focus on:

- Articles of immediate importance and usefulness
- Surveys of the research
- Editorial comment to spark debate
- A forum for question/answer and the sharing of ideas

The first issue will be available May 1984. A special introductory issue is *available free upon request. This special preview issue is 24 pages in length and contains subscription information for the regular issues.*

Coming Soon— Computing in the Social Studies Classroom

This new ICCE booklet discusses the roles of educational computing that apply to social studies, how to locate and evaluate computing materials for social studies as well as the role of social studies classroom computing within a school's instructional program. The authors are Allen Glenn, Professor of Curriculum and Instruction at the University of Minnesota; and Don Rawitsch, Director of User Services for the Minnesota Educational Computing Consortium (MECC). Available June 15, 1984. 48 pages, $3.50.

Preview Guide Now Available from ICCE

The *1984 Educational Software Preview Guide* is a list of favorably reviewed instructional microcomputer software for K-12 classroom use. It is designed to assist educators in locating software for preview; it is not intended to endorse these products for purchase without examination.

The *Guide* was developed by the Educational Software Evaluation Consortium, representing 27 organizations involved in computer education throughout North America. The selection of titles in this guide was based on critical evaluations conducted by the participating organizations and on discussion among the Educational Software Evaluation Consortium representatives who attended the 1984 California Software Evaluation Forum. Reviews published in professional journals were also considered. 8½x11, 50 pages, $5.

Logo in the Classroom

Logo in the Classroom is a collection of activities designed by Shirley Torgerson, an inservice and classroom teacher who takes Logo beyond "playing Turtle" and integrates it into the elementary curriculum. Based upon in-class experiences and expanded with suggestions from other Logo-using teachers, 20 complete lesson plans have been written for use in single-computer classrooms at the 3-6 grade level. The session guidelines encourage group-oriented activities as well as individual student exploration. Logo dialects covered are Terrapin (MIT) for Apple and Commodore 64, LCSI for Apple and Atari, Texas Instruments Logo and Radio Shack Logo. 8½ x 11, 202 pages, $13.

Some Special Offers from ICCE

ICCE Mini-subscription. A special 4 issue for $5 mini-subscription is available to teachers leading workshops, inservice or preservice courses of at least 15 people. Not a regular ICCE Membership, this 4 for $5 offer includes *either*:

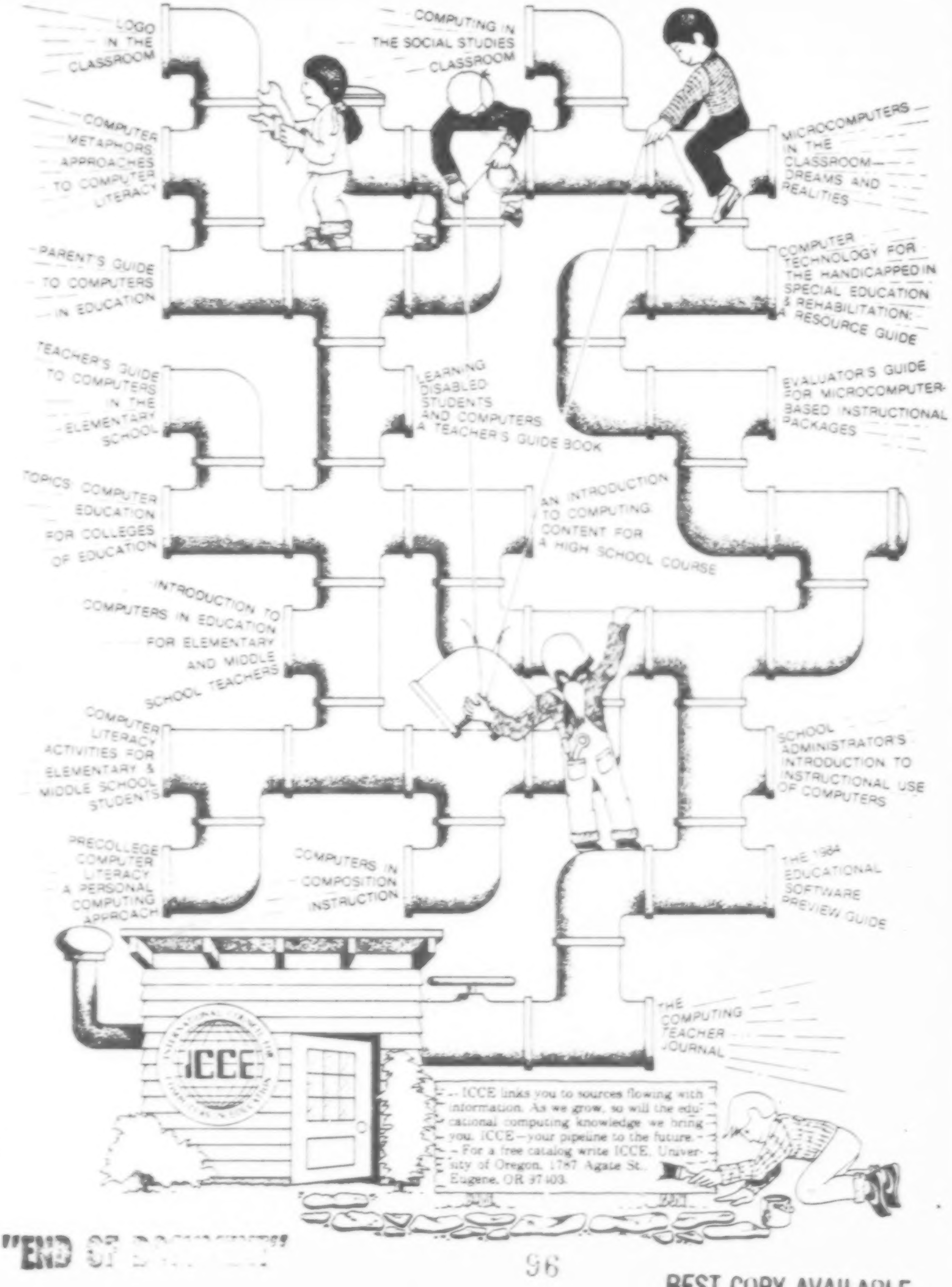
4 back issues received together *or*
2 back issues and the latest issue printed, received together with the next issue printed mailed to you.

Orders with fewer than 15 names going to one address cannot be processed.

ICCE Materials—Save $21.50! ICCE offers you a $96.50 value for only $75. Receive one copy of every booklet ICCE publishes and 9 issues of *The Computing Teacher* journal. That's 15 booklets and one year of TCT for $75!

For orders, write the International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403. (503) 686-4414.

BEST COPY AVAILABLE


LOGO IN THE CLASSROOM
COMPUTING IN THE SOCIAL STUDIES CLASSROOM
COMPUTER METAPHORS: APPROACHES TO COMPUTER LITERACY
MICROCOMPUTERS IN THE CLASSROOM—DREAMS AND REALITIES
PARENT'S GUIDE TO COMPUTERS IN EDUCATION
COMPUTER TECHNOLOGY FOR THE HANDICAPPED IN SPECIAL EDUCATION & REHABILITATION: A RESOURCE GUIDE
TEACHER'S GUIDE TO COMPUTERS IN THE ELEMENTARY SCHOOL
LEARNING DISABLED STUDENTS AND COMPUTERS: A TEACHER'S GUIDE BOOK
EVALUATOR'S GUIDE FOR MICROCOMPUTER-BASED INSTRUCTIONAL PACKAGES
TOPICS: COMPUTER EDUCATION FOR COLLEGES OF EDUCATION
AN INTRODUCTION TO COMPUTING: CONTENT FOR A HIGH SCHOOL COURSE
INTRODUCTION TO COMPUTERS IN EDUCATION FOR ELEMENTARY AND MIDDLE SCHOOL TEACHERS
COMPUTER LITERACY ACTIVITIES FOR ELEMENTARY & MIDDLE SCHOOL STUDENTS
SCHOOL ADMINISTRATOR'S INTRODUCTION TO INSTRUCTIONAL USE OF COMPUTERS
PRECOLLEGE COMPUTER LITERACY: A PERSONAL COMPUTING APPROACH
COMPUTERS IN COMPOSITION INSTRUCTION
THE 1984 EDUCATIONAL SOFTWARE PREVIEW GUIDE
THE COMPUTING TEACHER JOURNAL
ICCE

ICCE links you to sources flowing with information. As we grow, so will the educational computing knowledge we bring you. ICCE—your pipeline to the future.

For a free catalog write ICCE, University of Oregon, 1787 Agate St., Eugene, OR 97403.

"END OF DOCUMENT"

BEST COPY AVAILABLE